THE MESONH USER'S GUIDE

MASDEV4_7 version

April 2, 2008

Contents

Chapter 1

Introduction

MESONH is an atmospheric simulation model which can be run in very different conditions. Its capacities are completely described in the scientific documentation of the model, where the model equations and the whole parametrizations' set are given.

In this book, is given the necessary information to perform numerical experiments, using the MESONH atmospheric system, from the generation of the initial fields to the visualization of the results. It will help the user to correctly manipulate all the UNIX scripts, source file management, to prepare or exploit a numerical experiment made by this model. It is intended to give a clear presentation of how to work with the model for a new user and to be a reference guide for a more advanced one. For this reason, you will find illustrative examples in each chapter of this book, in order to have practical applications.

From a technical point of view, no special knowledge are required to perform a simulation. It is useful to specify the main characteristics of the model:

- The sources are written in **standard Fortran 90** (Metcalf and Reid 1993)
- The source file management is perform by the **standard UNIX file manager RCS**
- The procedures are written in **standard UNIX Korn-Shell**
- The generalized use of **dynamic memory allocation** avoids repeated compilation of the source and the free parameters are set by the user through namelist files

A sequence of elementary actions are to be done to realize a complete numerical experiment, as illustrated by Figure 1.1.

The first part is used to modify the Fortran sources of the MESONH library or to add new Fortran sources in order to adapt the model to the user's wishes. Nevertheless, because Meso-NH is able to perform a lot of actions without any modifications of the Fortran sources, we will end the presentation of the book with this first part, when the new user will know how to manipulate Meso-NH as it stands. The second part corresponds to the generation of the

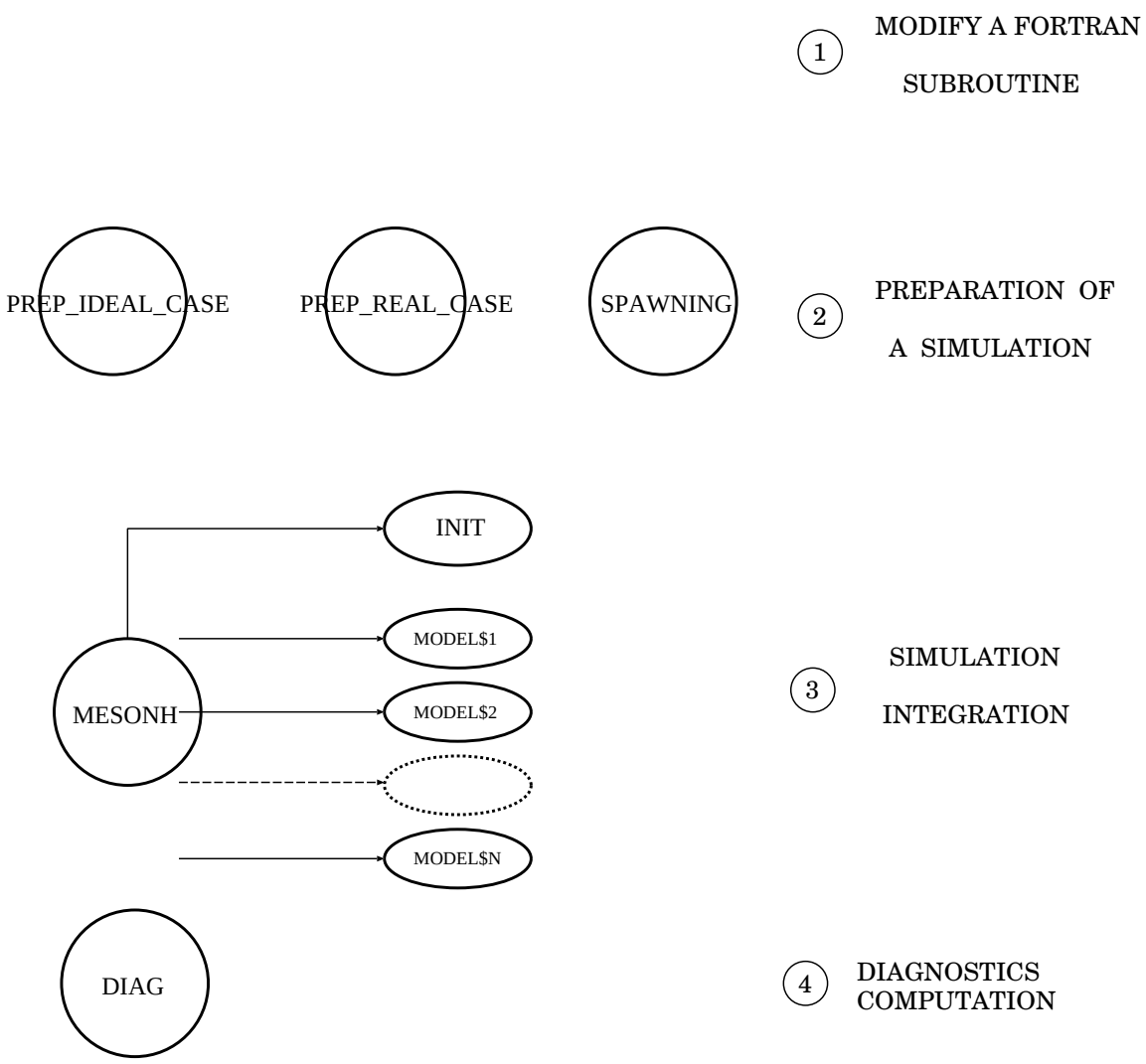


Figure 1.1: General algorithm for a complete numerical experiment

initial MESONH files, which contain the values of the prognostic variables of the model. This is realized by 3 differents programs, which correspond to the preparation either of an ideal case study (idealized atmosphere and orography), either of a real case study (interpolation from a Larger Scale operationnal model to the finer mesh MESONH model) or of a nested simulation (interpolation from the Larger Scale MESONH model to a new finer mesh MESONH model). The third part is the temporal integration of the model, starting from the initial values read in the initial MESONH files. During the integration, the model outputs are stored in MESONH files identical to the initial file. In the fourth part, diagnostic variables can be computed after the simulation.

To perform these different parts, some Unix procedures have been developped and are described in this book. A lot of parameters are to be given:

- to the procedure itself as for instance in which directory the input MESONH files are to be searched
- to the fortran source to correctly configurate the model as for instance the model time step

The presentation of these procedures is realized in Chapter 2, because they are fondamental tools to correctly manipulate the Meso-NH atmospheric simulation system. We will delay the presentation of how to modify the fortran source to the Chapter 7, where the source manager is presented in detail.

We will present in Chapter 3, how to generate initial conditions for a Meso-NH simulation. The fields are stored in a file, whose structure is described in the first Annexe. This file contains the values of all prognostic and some diagnostic fields in binary form. This file type will be called a FM-file in the following. We also describe the file which contain the free-parameters of the model under the form of a set of Fortran 90 namelists. We choose an academic case (i.e. simplified orography and atmospheric conditions) for the generation of initial conditions, because it makes easier the presentation, because less models and file types are involved. The real case presentation is delayed to Chapter 5.

In Chapter 4, the degrees of freedom of the model itself, are presented. We also recall where informations are read either to start the simulation (FM-file) or to pilot the MESONH model from a Larger Scale Model.

In Chapter 5, we present in the same way, the real case study and its own degrees of freedom. Several programs are involved to generate the initial MESONH FM-file, used either in a mono-model simulation or in a multi-model simulation with grid-nesting technics.

The Chapter 6 lists the diagnostic variables which can be obtained after a simulation.

In Chapter 7, the file manager is presented and we show how to modify or add Fortran sources to the MESONH sources. We also describe how to merge these new elements with the Meso-NH sources.

In Annexe 1, the FM-file structure is described in details.

Chapter 2

The MESONH procedures (except prepsource) :

2.1 Presentation

2.1.1 What can you do with the MESONH procedures ?

The MESONH procedures provide a complete UNIX shell script which allows the MESONH user to compile a subroutine, to prepare a numerical simulation, to run the model, to compute diagnostic fields...

This Chapter will therefore present how to form a job built with one or more elementary steps. It should be noted that a very modular structure has been adopted for the procedures and thus, a job performing more than one elementary step is only the assembling of the different scripts performing each elementary step. A script is a UNIX procedure which:

- collects informations from different sources (environment variables, the content of one or more files containing Fortran subroutines or lists of variables...)
- puts together all theses informations in a single file (named outexample for the script example)
- executes the job included in this file.

The elementary scripts (fig.2.1) can perform the following actions:

- modify your binary library by compiling a subroutine which is added or modified by comparison with the master library, including the Bugfix library or not (step **prepsource**), the presentation of this step is described in chapter 7.
- perform either the preparation of an initial MESONH file either the time-dependent simulation or **any main program present in the master library** (step **prepmodel**)
- execute your complete job (step **tosupc**).

- ask for operational datas from ECMWF (step **extractecmwf**) or from Meteo-France (step **extractarpege**)

This user's guide is not the only way to get informations on the procedures, the MESONH user can get part of them, directly on its screen. He can get brief informations on the `script` procedure by making :

```
script ?
```

He will then get a list of information: purpose, synopsis, list of input parameters and input files. It also contains some words on the calling scripts and the called scripts. He can also access a man type information by:

```
man script
```

It contains an extended information about the procedure, including especially a more or less detailed description of the method used in the script. Some informations are similar to those present in the on-line help.

As previously quoted, each script, except `tosupc`, creates a file called out(script). This file contains a set of executable UNIX commands that can be executed on the remote_host (Fujitsu, NEC or UNIX workstation), and every elementary step adds a new part to this file. In this picture and later on, we will refer to "local host" as the machine on which the preparative script is run. The output file (i.e. out(script)) is executed on the "remote host". Nevertheless, the "remote host" may be identical to the "local host" if desired.

In the first chapters, we will use the model as it stands. The compilation of the MESONH sources has been previously done and the binaries objects generated during this compilation step have been collected in binary libraries localized on the remote host. The different binaries are loaded by the procedure outprep in order to obtain the right executable binary (ABS$MAINPROG.exe) in the work directory on the remote host where the model executions are performed. The chapter 7 will explain these steps.

The different localizations and information transfers are illustrated in figure 2.2. The vertical line separates what happens in the local host from the actions performed in the remote host.

The first part of an experiment consists in the preparation of a file named outscript, which is composed of a list a UNIX orders and a copy of the pertinent namelists. This file is transfered in the remote host, where it is executed, this job is self-governing and is able to extract the required file either from the remote host or from a storage machine linked to it. The different outputs (listings, FM synchronic files, FM diachronic files...) are disposed at the rigth destinations during still the same job included in the outscript file (local host, remote host, storage machine...).

Before the extensive presentation of the different steps of the **prepmodel** and **tosupc**

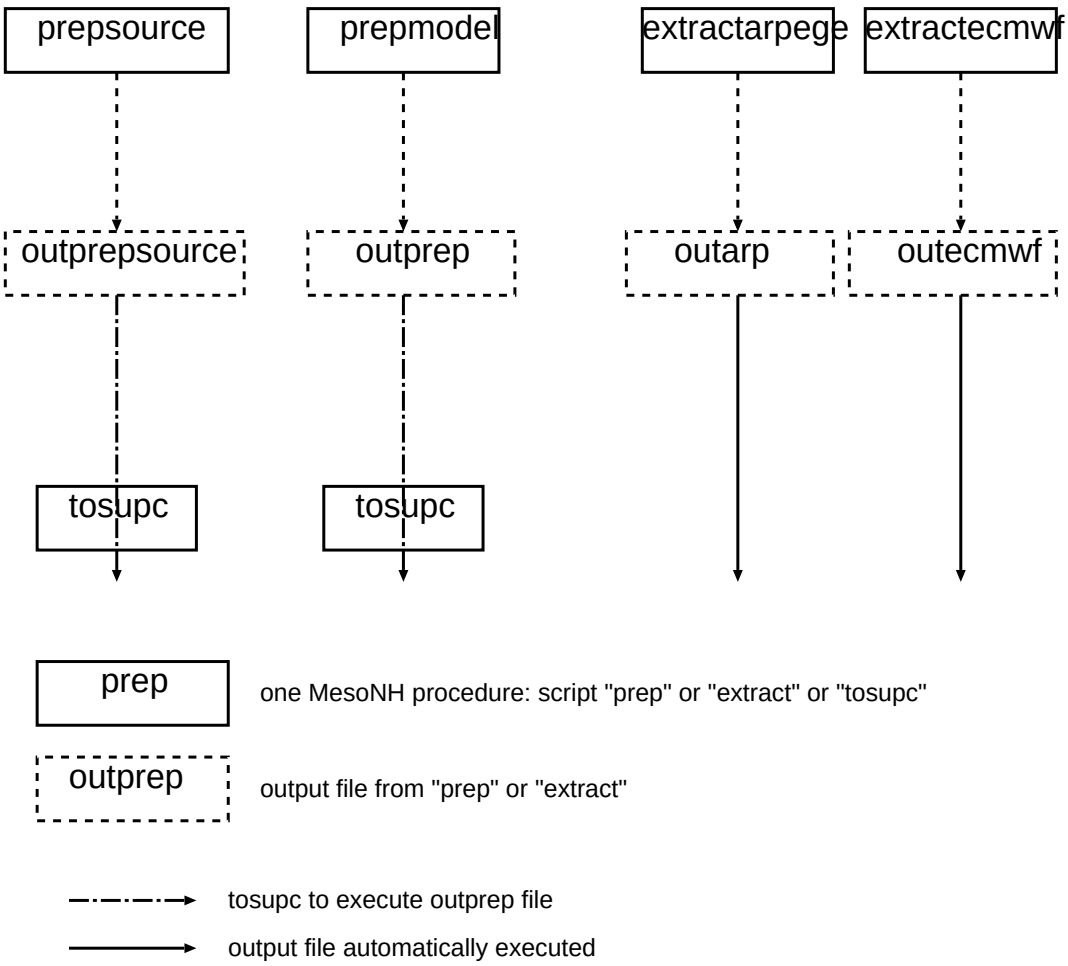


Figure 2.1: MESONH procedures

procedures, we briefly describe the different ways to get informations: either by environment variables which must be initialized before running the procedures or as input parameters for the procedures themselves.

2.1.2 The environment variables

The environment variables are necessary for the UNIX procedures to recover the right paths where the informations must be taken out. The environment variables have to be set up in the .profile of the user on the local host and are listed below:

1. ENVIRONMENT: indicates how the MESONH scripts will be run. There are 4 possible values:

   - INTERACTIVE: the scripts, which prepare the job that will be executed later on the remote host, are called during an interactive session on the local host. Full screen

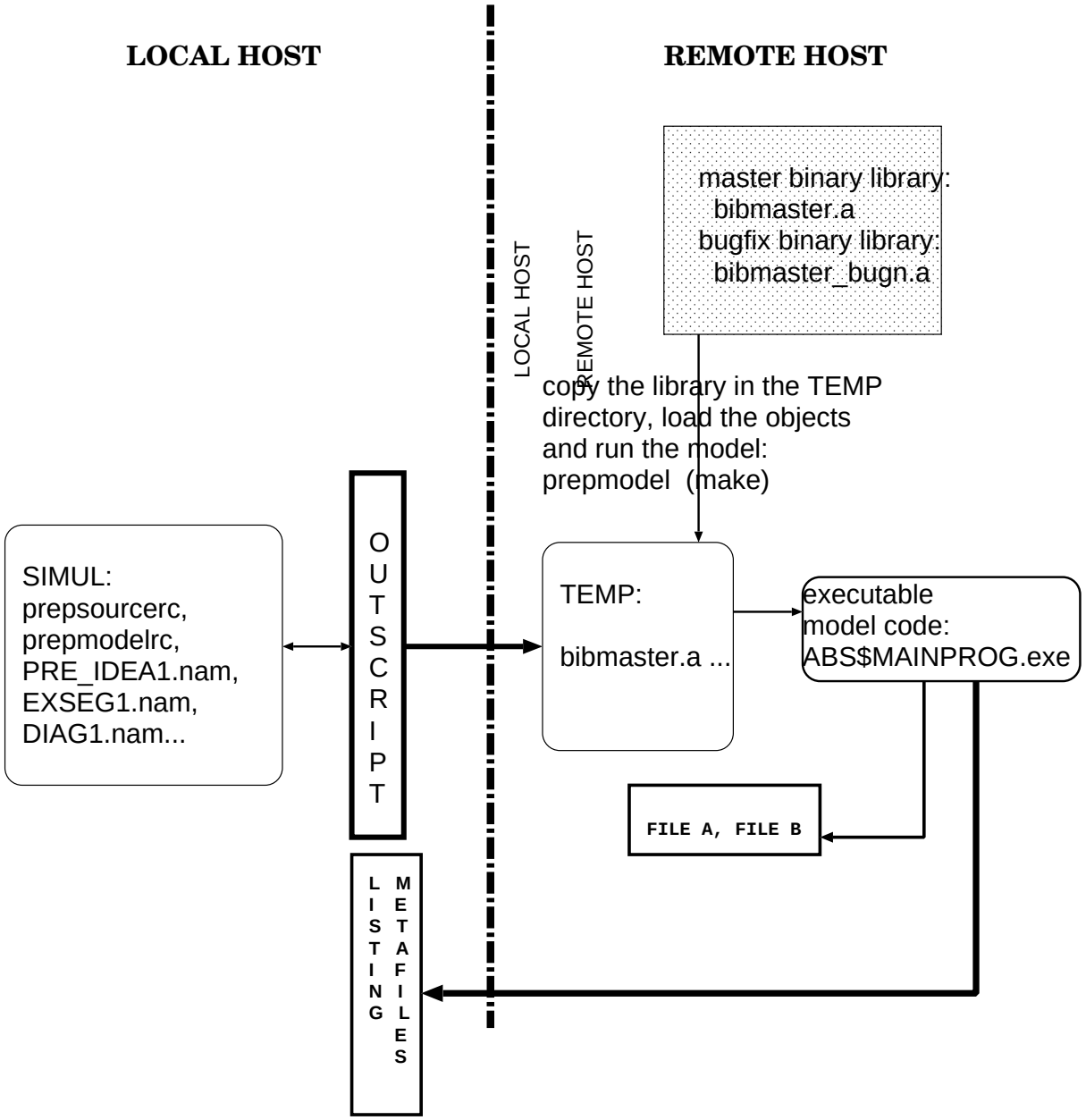


Figure 2.2: The different locations of the informations

initialization is allowed and some questions may be asked to the user. INPUT = files + keyboard and OUTPUT = screen.

- SILENTINTERACTIVE: the scripts are called in an interactive session but no full screen initialization and no questions are asked. INPUT = files and OUTPUT = screen.
- BATCH: the scripts are called in batch mode. INPUT = files and OUTPUT = file.
- CRON: the scripts are called with `crontab`. INPUT = files and OUTPUT = file.

2. DEBUGSCRIPT: switches command echos on or off, for an easy script debugging: ON or OFF (this debugging level reports to the procedures and not to the Fortran 90 subroutines compilation)

3. PATH: the pathname to find the Meso-NH procedures must be added to PATH. For example : PATH=$PATH:/mesonh/procedures

4. MANPATH: the path to find the `man` files must be added

5. EDITOR: this is the ascii file editor, often set to `vi`.

6. SIMUL: absolute name (path+ directory name) for the simulation directory on which the MESONH procedures will work (it contains extracted source files, namelist files, specific files containing personal default values used by the procedures)

2.1.3 The input parameters of a MESONH script

The meaning of the control parameters needed by every main MESONH script is described in the next section. We only present, here, the different ways to set up these parameters:

- They can be introduced via variables stored in input files named `scriptrc`(`script` stands for the name of any procedure and `rc` is the suffix). If `scriptrc` is not provided in the SIMUL directory, then a default file in the Meso-NH root directory ($MESONH/ procedures) will be searched. (These files are strongly recommended because they are obligatory in a non-interactive session and used to provide own defaults for an interactive session). To create your own version of the `scriptrc` file, you may do:

  ```
  cp $MESONH/procedures/scriptrc $SIMUL/scriptrc
  ```

- They may be again modified when calling a procedure with an explicit argument list:

  ```
  script VAR=VALUE
  ```

  (`script ?` will give you the list of the "VAR" accepted by the script.)

- They may also be changed during the full screen initialization step, when ENVIRONMENT is set to INTERACTIVE.

- Input files can be used by some scripts (`prepsource`, e.g.) in SILENTINTERACTIVE. These input files have the general name `in(script)` and do not concern the input parameters themselves, but they contain input that is read by `script`. Thus, whereas in INTERACTIVE `script` would ask the user for information, in SILENTINTERACTIVE, it reads the data from file `inscript`.

2.2 A complete Meso-NH work session

For the following, we recall that "local host" stands for the machine on which the preparative scripts are running and "remote host" stands for the machine on which the Meso-NH job is actually executed.

2.2.1 Source extraction and compilation (`prepsource`)

This part will be described later in chapter 7 because it is only useful when you do not want to use the MESONH model as it stands.

2.2.2 Running the model, Preparing an initial file or Computing diagnoctic fields (`prepmodel`)

The procedure `prepmodel` prepares a Meso-NH job to run the compiled code with a given main program. For most of them (PREP_IDEAL_CASE, PREP_REAL_CASE, PREP_PGD, PREP_NEST_PGD, ZOMM_PGD, MODEL, DIAG, SPAWNING), the namelist file name ($NAMELIST-FILE) and the variables to get in ($LISTGET) are then automatically initialized (see tab.2.1). We can distinguish six steps, which will be executed on the remote host (see figure 2.2):

1. it performs the load step if the binary libraries of object files have been modified and saves the executable ABS$MAINPROG.exe in $workdir, or it only copies it from $workdir to $TEMP.

2. the input files, read in the namelist file, are search and copy to $TEMP/execdir. If the files are got from the storage-file machine, they are also copied on $workdir/$INDIR directory to prevent next transfer. The next time the file will be got from this directory if it is still here.

3. it executes ABS$MAINPROG.exe (i.e. the fortran code, in $TEMP/execdir)

4. it saves the output files on remote host ($workdir/$OUTDIR or $HOME/$OUTDIR) or on the storage-file machine ($HOME/$OUTDIR) system. In the latter case it also copies the output files on the $workdir/$OUTDIR directory to prevent next transfers.

5. it verifies if output files are still present and were not yet saved. If necessary, it saves them on remote host or the storage-file machine.

6. it sends to the local host a listing including all the echoes of the UNIX orders and a copy of the outputs directly generated by the Fortran program.

Again, control variables are needed. These parameters are set up in a file `prepmodelrc` stored on $SIMUL and can be modified in an interactive session. We now give their list:

- MKFNAME= file which contains the makefile
- BIBMASTER= master library's name
- BIBBUGFIX= bugfix library's name or 0 (none)
- BIBUSER= user library's name (starting at $HOME) or 0 (none)
- DEBUG= option of the run (=run normal execution or =debug for debugging option)
- LOAD_OPT= loading options added to the default ones, they must be between double quote (example:"-lnag -m")
- MAINPROG= one of the MesoNH programs (PREP_IDEAL_CASE, PREP_REAL_CASE, PREP_PGD, PREP_NEST_PGD, ZOMM_PGD, MODEL, DIAG, SPAWNING),
  or another name to run any program developped in the Mesonh environment.
- NAMELISTFILE= is set to "default" if MAINPROG is one of the MesoNH programs (see tab.2.1),
  else it must be set to the filename which contains the namelists for the $MAINPROG program.
- LISTGET= is set to "default" if MAINPROG is one of the MesoNH programs,
  else it must be set to the list of the variables which are extracted from the $NAMELIST-FILE to find the name(s) of the input FM files
- INHOST= input mesonh files are on execution (remote host) or storage-file machine: supc/archiv
- INDIR= directory of the input mesonh files:
  starting at $HOME (0=$HOME) for storage-file machine
  home/... = starting at $HOME for remote host storage
  work/... = starting at $workdir for remote host storage
- INLOGIN= user,on the $INHOST machine, of the input mesonh files
  (0= the $LOGNAME on the remote host)
- OUTHOST= output mesonh files will be on execution (remote host), storage-file machine or workstation: supc/archiv/workstation_name
- OUTDIR= directory of the output mesonh files: as INDIR

MAINPROG	Namelist file	LISTGET	OUTSCRIPT
CONVLFI	CONVLFI1.nam	CMNHFILE	outprepconv
DIAG	DIAG1.nam	YINIFILE	outprepdiag
MODEL	EXSEG1.nam	CINIFILE	outprepmodel
PREP_IDEAL_CASE	PRE_IDEA1.nam	CCPLFILE CPGD_FILE	outprepideal
PREP_PGD	PRE_PGD1.nam	YZS YCLAY YSAND YSST	outprep_pgd
PREP_NEST_PGD	PRE_NEST_PGD1.nam	YPGD1 ... ... YPGD8	outprepnest
PREP_REAL_CASE	PRE_REAL1.nam	HATMFILE HPGDFILE	outprepreal
SPAWNING	SPAWN1.nam	CINIFILE YDOMAIN YDADINIFILE YDADSPAFILE	outprepspawn
ZOOM_PGD	PRE_ZOOM1.nam	CPGDFILE	outprepzoom
MY_PROG	MY_NAM1.nam NAMELISTFILE=MY_NAM	LISTGET=' '	outprepmodel

Table 2.1: The different MesoNH programs and Namelist files associated

- SUBMIT_NEXTJOBS= name of one or several jobs to submit to remote host at the end of this job (pathnames relative to the current $SIMUL, `tosupcrc` must be present in the same directory): 0/'filename'.
- LOCAL_NEXTJOBS= name of jobs to execute on local host at the end of this job with "at" procedure (pathnames relative to the current $SIMUL or absolute): 0/'dirlocal filename'.

The last two control parameters allow to link several jobs together, the first one is executed on the remote host (output of `prepmodel`), the other ones are either executed on the remote host (e.g. a model simulation after the preparation of the initial file) or on the local host (e.g. a trace session after computation of diagnostics).

2.2.3 Send your job for execution (`tosupc`)

Output scripts prepared before (outprepsource or outprep`prog`) with shell commands have to be executed on the remote host. Here we describe what tools are used to submit and execute the output script.

The procedure `tosupc` allows either to send the job to the remote host and to put it into its wait queue, or to run the job on the local host. Thus, `tosupc` runs a job in batch on the remote host or in batch or interactive on the local host, depending on the control parameters described below. Another way to work is to execute the job interactively on the remote job. It can be put on the remote host by ftp or rcp and then executed interactively, if it is possible to

do it (it is safier to work on the $tmp of the remote host). Just execute on this machine the following order: `outscript`.

Here are the control parameters for `tosupc` (file `tosupcrc`):

- JOBFILE = filename which contains the script to be run
- JOBNAME = job name on the remote host (JOBFILE by default)
- TIME = time in seconds class
- MEM = memory with unit (default is mbytes except mwords for Cray)
- TPN = number of processors on each node
- NBP = mono or multi-node job
- MAIL = is sent at the beginning, or the end, or both, of the job

`tosupc` calls another UNIX procedure which effectively transfers the ouput script file to the remote host, but this step is very machine dependant (local host and remote host). Anyway, limits for the duration, values of memory, processors and nodes of the job are typical informations which must be provided to correctly configurate this step.

Chapter 3

Prepare an initial MESONH file for an ideal case

3.1 Overview of PREP_IDEAL_CASE functionalities

The "PREP_IDEAL_CASE" program prepares a MESONH file, that contains all the parameters and fields necessary for the execution of the MESONH model. Specifically, the grid parameters, the initial fields and the geophysical fields are included in this file. It is possible using this program to generate idealized fields defined by few parameters. This is useful for :

- debugging purpose
- a simulation starting from an idealized initial field.

The generated initial conditions are produced analytically, leading to quasi-1D fields or 3D fields or a single profile build with either:

- layers of constant Brunt-Vaisala frequency, shear and humidity or
- a Radiosounding and ideal surface fields
- a Radiosounding and real physiographic fields
- a Radiosounding and real and ideal surface fields at the same time

For these latter cases, the initial fields may be hydrostatically or geostrophically balanced or not. For these fields to satisfy the anelastic constraint, a final correction is applied to them.

The interaction between the PREP_IDEAL_CASE program and the user is made through the PRE_IDEA1.nam file. The degrees of freedom are collected in a set of namelists, read by this program.

3.2 The input: the PRE_IDEA1.nam file

It is made of two parts :

- A namelist-part with directives for the preparation of an idealized case (always present). The order of namelists is free and unset namelists can be ommited.
- A free-formatted part describing a vertical profile of n layers of constant moist Brunt-Vaisala frequency or a radiosounding and sometimes the explicit list of the heights of the vertical levels. This part can be present or absent in the other cases.

To initialize a simulation with a radiosounding and real terrain conditions, it is necessary to perform the PREP_PGD program (see next chapter) to create a MESO-NH physiographic data file. This data file contains the orography and the physiographic data fields (related to the soil scheme). It is also possible to perform a complete ideal case with ideal orography and non trivial surface conditions. The user can combine the two possibilities with switches included in the namelist NAM_REAL_PGD and initialize a simulation with a real orography and idealized homogeneous surface fields. If a PREP_PGD file is specified and if the switches in namelist NAM_REAL_PGD are set to FALSE, homogeneous values can be imposed by the user in namelists from the externalized surface facility PGD (namelists NAM_COVER and NAM_ISBA), else the PREP_PGD fields are taken into account.

The namelists are then list alphabetically.

3.2.1 Namelist NAM_BLANK (available variables)

see Perform a MESONH simulation for description

3.2.2 Namelist NAM_CH_MNHCn_PRE (init. chemistry scalar variables)

If you initialize MNH-C using PREP_IDEAL_CASE, use the following namelist variables:

Fortran name	Fortran type	default value
LCH_INIT_FIELD	logical	FALSE
LORILAM	logical	FALSE
CCHEM_INPUT_FILE	80 characters	MNHC.input

- LCH_INIT_FIELD: switch to activate initialization subroutine CH_INIT_FIELD.
- CCHEM_INPUT_FILE: name of the general purpose input file for initialization.
- LORILAM: switch to activate initialization chemical aerosol (only if LCH_INIT_FIELD=T).

3.2.3 Namelist NAM_CONF_PRE (configuration variables)

Fortran name	Fortran type	default value
LCARTESIAN	logical	TRUE
LPACK	logical	TRUE
CEQNSYS	3 characters	'DUR'
NVERB	integer	5
CIDEAL	4 characters	'CSTN'
CZS	4 characters	FLAT
LBOUSS	logical	FALSE
LPERTURB	logical	FALSE
LFORCING	logical	FALSE

- LCARTESIAN : Switch for cartesian geometry
  - .TRUE. for cartesian geometry
  - .FALSE. for conformal projection
- LPACK : Switch to compress FM file for 1D or 2D version.
- CEQNSYS : Equation system resolved by the MESONH model
  - 'LHE' Lipps and HEmler anelastic system
  - 'DUR' approximated form of the DURran version of the anelastic sytem
  - 'MAE' classical Modified Anelastic Equations but with not any approximation in the momentum equation
- NVERB : verbosity level
  - 0 for minimum of prints
  - 5 for intermediate level of prints
  - 10 for maximum of prints.

  If $CSURF$="EXTE" in namelist NAM_GRn_PRE , NVERB=10 prints two LaTeXfiles containing the initialisation of surface scheme variables for each type of surface cover (in french or in english).
- CIDEAL : kind of idealized fields
  - 'CSTN' : Constant moist Brunt Vaisala frequency case
  - 'RSOU' : radiosounding case
- CZS : orography selector The formulae are given below in the description of the namelist NAM_GRIDH_PRE.

  - 'FLAT' : constant XHMAX orography (zero by default)
  - 'SINE' : sine-shaped orography
  - 'BELL' : bell-shaped orography
  - 'DATA': discretized orography. The data describing the orography are given in the free format part. Only the orography corresponding to the computational domain must be provided in free format. For 3D orography, data are read like if it was a map (the first line is the Northern border and the first data is the North-West corner) with one line per Y-axis increment.

- LBOUSS : Switch for a Boussinesq version.
  - .TRUE. The reference anelastic state is $\theta_{ref} = cte = \theta_{ref}(z = 0)$ and $\rho_{ref} = cte = \rho_{ref}(z = 0)$. In this case, the stratification is taken into account in the Meso-NH model in the flottability term. The typical length, on which this stratification varies, is much greater than the domain heigth and the θ_{ref} variation can be therefore neglected.
  - .FALSE. The reference anelastic state varies with the altitude.

- LPERTURB : Switch to add a perturbation on the initially horizontally homogeneous fields. This perturbation is not balanced.

  3 perturbation types are implemented in the routine *set_perturb.f90* :
  - a spherical perturbation on the dry potential temperature and the moisture fields typical for convection initialization.
  - a perturbation on the horizontal components of the wind derived from a streamfunction, typical for large scale studies. This insures the wind against becoming divergent.
  - a perturbation on the dry potential temperature field at the first mass level near the ground, corresponding to a white noise (uniform amplitude in the spectral space). This type of perturbation is used in Large Eddy Simulations' initialization.

  When .TRUE., the parameters for the precise definition of the perturbation can be set in the namelist NAM_PERT_PRE or sometimes must be modified directly in the subroutine *set_perturb.f90*

- LFORCING : Switch to specify forcing sources. When .TRUE., the precise definition of the forcing is set in the free-format part of PRE_IDEA1.nam (see 3.3.4).

3.2.4 Namelist NAM_CONFn (configuration variables for modeln)

Fortran name	Fortran type	default value
LUSERV	logical	TRUE
LUSERC	logical	FALSE
LUSERI	logical	FALSE
NSV_USER	integer	0

(see 3.3.2 for more details for these cases)

- LUSERV : Switch to write r_v (vapor mixing ratio) in initial file. It is reset to .TRUE. when CIDEAL ='RSOU' or 'CSTN'. This has been done in order to avoid to treat the dry case as a particular case but as a moist case with humidity equal to 0.
- LUSERC : Switch to write r_c (cloud mixing ratio) in initial file. This case is only allowed when CIDEAL ='RSOU' (radiosounding case) and KIND='PUVTHDMR' or KIND='ZUVTHLMR'
- LUSERI : Switch to write r_i (ice mixing ratio) in initial file. This case is only allowed when CIDEAL ='RSOU' (radiosounding case) and KIND='PUVTHDMR'
- NSV_USER : number of scalar variables Note that if NSV_USER is different from 0, the Scalar Variables are initialized to 0 by the program

3.2.5 Namelist NAM_DIMn_PRE (contains dimensions)

Fortran name	Fortran type	default value
NIMAX	integer	10
NJMAX	integer	10

- NIMAX : number of mass points in x-direction of the initial file is $NIMAX + 2JPHEXT$ ($JPHEXT$ corresponds to the number of marginal points in the horizontal directions and is fixed to 1 for the present Meso-NH version)
- NJMAX : number of mass points in y-direction of the physical domain. The total size of the array written in the initial file is $NJMAX + 2JPHEXT$

3.2.6 Namelist NAM_DUST_PRE (init. dust scalar variables)

If you initialize aerosol passive dust during PREP_IDEAL_CASE, use the following namelist variables:

Fortran name	Fortran type	default value
LDUST	logical	FALSE

- LDUST: switch to activate initialization of passive dust (3 modes).

3.2.7 Namelist NAM_DYNn_PRE (pressure solver)

Fortran name	Fortran type	default value
CPRESOPT	5 characters	'CRESI'
NITR	integer	4
XRELAX	real	1.

- CPRESOPT : gives the type of pressure solver used for the elliptic equation ('RICHA', 'CGRAD', 'CRESI'). This equation is solved in order to ensure the anelastic constraint for the initial wind field. Note that the solver is applied even for the FLAT case when the Earth spericity is taken into account.
- NITR : number of iterations used for the elliptic equation resolution (solver = "CPRESOPT").
- XRELAX : relaxation factor used by the Richardson method (CPRESOPT = "RICHA").

3.2.8 Namelist NAM_GRID_PRE (grid definition)

Fortran name	Fortran type	default value
XLON0	real	0.
XLAT0	real	60.
XBETA	real	0.
XRPK	real	1.
XLONORI	real	350.
XLATORI	real	37.

- XLON0 : reference longitude for conformal projection (if LCARTESIAN =.TRUE. this value can be usefull to compute local solar time)
- XLAT0 : reference latitude for conformal projection and cartesian plane
- XBETA : rotation angle for conformal projection and cartesian plane
- XRPK : cone factor for the projection (not used if LCARTESIAN =.TRUE.):
  - XRPK=1: polar stereographic projection from south pole
  - 1>XRPK>0: Lambert projection from south pole
  - XRPK=0: Mercator projection from earth center
  - -1<XRPK<0: Lambert projection from north pole
  - XRPK=-1: polar stereographic projection from north pole
- XLONORI : Longitude (in degrees) of the origine point (not used if LCARTESIAN =.TRUE.). This point is the mass point of conformal coordinates (x=0,y=0) of the Meso-NH grids.
- XLATORI : Latitude (in degrees) of the origine point (not used if LCARTESIAN =.TRUE.)

3.2.9 Namelist NAM_GRIDH_PRE (horizontal grid definition)

Fortran name	Fortran type	default value
XDELTAX	real	5000.
XDELTAY	real	5000.
XHMAX	real	300. / 0.
NEXPX	integer	3
NEXPY	integer	1
XAX	real	10000.
XAY	real	10000.
NIZS	integer	5
NJZS	integer	5

- XDELTAX : mesh length (in meters) in x-direction on the conformal or cartesian plane. It is not used if you read informations in a Meso-NH constant file (PGD_FILE).
- XDELTAY : mesh length (in meters) in y-direction on the conformal or cartesian plane. It is not used if you read informations in a Meso-NH constant file (PGD_FILE).
- XHMAX[1] : Maximum height (in meters) h_{max} for orography (case CZS $\neq$ 'FLAT') or ground level for flat orography
- NEXPX : Exponent exp_x for orography in case of CZS='SINE'
- NEXPY : Exponent exp_y for orography in case of CZS='SINE'
- XAX : Widths (in meters) a_x along x for orography in case CZS='BELL'

$$z_s(\hat{x},\hat{y}) = \frac{h_{max}}{\left(1+\left(\frac{\hat{x}-NIZS*XDELTAX}{XAX}\right)^2+\left(\frac{\hat{y}-NJZS*XDELTAX}{XAY}\right)^2\right)^{1.5}}$$

  in the three-dimensional case.

$$z_s(\hat{x}) = \frac{h_{max}}{1+\left(\frac{\hat{x}-NIZS*XDELTAX}{XAX}\right)^2}$$

  in the two-dimensional case.

- XAY : Widths (in meters) a_y along y for orography in case CZS='BELL'
- NIZS : Localization in x-direction of the mountain center in the case CZS ='BELL'. ($x_s = NIZS * XDELTAX$) It refers to a vertical velocity point at the ground (NIZS, NJZS)
- NJZS : Localization in y-direction of the mountain center in the case CZS ='BELL'. ($y_s = NJZS * XDELTAY$)

[1] default is 300. for mountain and 0 for flat orography

3.2.10 Namelist NAM_GRn_PRE (soil scheme choice)

Fortran name	Fortran type	default value
CSURF	4 characters	"NONE"

- CSURF : ground selector.
  - 'NONE' no soil scheme will be activated during the future MesoNH simulation, we therefore do not need any soil parameters. All the namelists of the externalized surface will be ignored.
  - 'EXTE' the externalized surface is used, allowing for its amazing large choice of surface schemes.

3.2.11 namelist NAM_LBCn_PRE (lateral boundary conditions)

Fortran name	Fortran type	default value
CLBCX	array(2 characters)	2*"CYCL"
CLBCY	array(2 characters)	2*"CYCL"

- CLBCX : represent the type of lateral boundary condition at the left and right boundaries along x (CLBCX(1) and CLBCX(2) respectively). Possible values are "CYCL", "OPEN", "WALL" for cyclic, open and rigid wall boundary conditions. It should be note that CLBCX(1) or CLBCY(1) refers to the lowest index values (IIB , IJB for X and Y directions) and CLBCX(2) or CLBCY(2) to the highest index values (IIE and IJE). Please note that :

$$CLBCo(1) = "CYCL" \Rightarrow CLBCo(2) = "CYCL"$$

  These boundaries conditions must correspond to the ones, which will be used for the Meso-NH run itself.

- CLBCY : array containing 2 elements: they represent the type of lateral boundary condition at the left and right boundaries along y (CLBCY(1) and CLBCY(2) respectively). They are strings of 4 characters.

3.2.12 Namelist NAM_LUNITn (logical unit names)

Fortran name	Fortran type	default value
CINIFILE	28 characters	'INIFILE'

- CINIFILE : name of the initial FM-file which will be produced by PREP_IDEAL_CASE, it will be used as initial file in a MESONH numerical simulation.

3.2.13 Namelist NAM_PERT_PRE (set analytical perturbations)

Fortran name	Fortran type	default value
CPERT_KIND	characters	'TH'
XAMPLITH	real	1.5
XAMPLIRV	real	0.0
XAMPLIUV	real	1.0834
XAMPLIWH	real	0.1
NKWH	integer	2
LSET_RHU	logical	TRUE
XCENTERZ	real	2000.
XRADX	real	10000.
XRADY	real	10000.
XRADZ	real	2000.

- CPERT_KIND: Defines the type of the perturbation
  - 'TH' the perturbation is on the thermodynamical fields (θ and r_v)
  - 'UV' the perturbation is on the horizontal wind fields (U and V)
  - 'WH' the perturbation is a white noise applied to θ
- XAMPLITH: Perturbation amplitude maximum for θ
- XAMPLIRV: Perturbation amplitude maximum for r_v
- XAMPLIUV: Perturbation amplitude maximum for U and V
- XAMPLIWH: Perturbation amplitude maximum for the normalized white noise
- NKWH: Upper level of the layer starting from the ground where the white noise is applied
- LSET_RHU: Conservation of the relative humidity
  - TRUE the relative humidity is conserved in the θ perturbation
  - FALSE the r_v perturbation is computed with the XAMPLIRV amplitude
- XCENTERZ: Height of the maximum of the θ perturbation
- XRADX: X-radius of the perturbation
- XRADY: Y-radius of the perturbation
- XRADZ: Z-radius of the perturbation

3.2.14 Namelist NAM_REAL_PGD (PGD file switches)

Fortran name	Fortran type	default value
CPGD_FILE	characters	' '
LREAD_ZS	logical	FALSE
LREAD_GROUND_PARAM	logical	FALSE

- CPGD_FILE : name of the physiographic data file containing the ground data fields. The file must be generated by the PRE_PGD program. **For a purely ideal case, the CPGD_FILE variable may be deleted from the namelist or set to its default value ' '. The horizontal grid will be read in the PGD file and therefore, the mesh increments XDELTAX and XDELTAY are no more used**.
- LREAD_GROUND_PARAM : switch to use or not the surface cover types (COVERnnn) and all other physiographic fields (except orographic ones) read in the PGD file.
  - .TRUE. to use the data read in the PGD_FILE
  - .FALSE. to use XUNIF_COVER idealized homogeneous values given in the namelist NAM_COVER (from the externalized surface) and scratch the PGD_FILE data
- LREAD_ZS : switch to use or not the orography parameters read in the PGD file.
  - .TRUE. to use the data read in the PGD_FILE
  - .FALSE. to use an idealized orography given in the namelist NAM_GRIDH_PRE and scratch the PGD_FILE data

3.2.15 Namelist NAM_SLEVE (smoothed orography for Sleve coordinate)

Fortran name	Fortran type	default value
NSLEVE	integer	12
XSMOOTH_ZS	real	XUNDEF

- NSLEVE : number of iteration for computation of smooth orography.
- XSMOOTH_ZS : optional uniform smooth orography.

3.2.16 Namelist NAM_VER_GRID (contains vertical grid definition)

There are three ways to compute the vertical grid, as in PREP_REAL_CASE:

1. constant grid mesh: only the number of levels NKMAX and the grid mesh sizes ZDZGRD and ZDZTOP are used. These must be equal. The type of grid YZGRID_TYPE is set to 'FUNCTN'.

2. two layers are defined, with constant stretching in each of these, the grid mesh sizes being given near the ground and at top of the model. It is possible that the top grid size is never reached, if the number of points is not enough for the prescribed stretchings. The type of grid YZGRID_TYPE is also set to 'FUNCTN'.

3. the levels are given by the user. The type of grid YZGRID_TYPE is set to 'MANUAL' in the namelist, and only the number of levels NKMAX is also used in it.

The variables of this namelist are:

Fortran name	Fortran type	default value
LTHINSHELL	logical	.FALSE.
NKMAX	integer	10
YZGRID_TYPE	6 characters	'FUNCTN'
ZDZGRD	real	300.
ZDZTOP	real	300.
ZZMAX_STRGRD	real	0.
ZSTRGRD	real	0.
ZSTRTOP	real	0.
LSLEVE	logical	FALSE
XLEN1	real	7500.
XLEN2	real	2500.

- LTHINSHELL : switch for the thinshell approximation (logical)
- NKMAX : number of points in z-direction of the required physical domain. The total size of the array written in initial file will be $NKMAX + 2JPVEXT$ ($JPVEXT$ is fixed to 1 for the present version of Meso-NH)
- YZGRID_TYPE : type of vertical grid definition:
  - 'FUNCTN': the vertical grid is given by a regular logarithmic function, whose variation is determined by the values of free parameters ZDZGRD, ZDZTOP, ZSTRGRD, ZSTRTOP, ZZMAX_STRGRD described below.
  - 'MANUAL': the levels are explicitly given in the free-formatted part by entering the heights of the different levels from the K=2 to K= KMAX + 2. Therefore, **you only enter KMAX +1 values**, because the level under the ground (i.e. K=1) is at the same distance from the ground (K=2) as the first level above the ground (K=3). Note also that the K= KMAX + 2 level represents the model top. In this case the free parameters (ZDZGRD, ZDZTOP, ZSTRGRD, ...) are not used
- ZDZGRD : mesh length in z-direction near the ground
- ZDZTOP : mesh length in z-direction near the top of the model

- ZZMAX_STRGRD : Altitude separating the two constant stretching layers
- ZSTRGRD : Constant imposed stretching (in %) in the lower layer (below ZZMAX_STRGRD)
- ZSTRTOP : Constant imposed stretching (in %) in the upper layer (above ZZMAX_STRGRD)
- LSLEVE : flag for Sleve vertical coordinate.
- XLEN1 : decay scale for smooth topography (in meters)
- XLEN2 : decay scale for smale-scale topography deviation (in meters)

3.2.17 Namelist NAM_VPROF_PRE (variables for CIDEAL ='CSTN' or 'RSOU')

Fortran name	Fortran type	default value
LGEOSBAL	logical	TRUE
CFUNU	3 characters	ZZZ
CFUNV	3 characters	ZZZ
CTYPELOC	6 characters	IJGRID
XLATLOC	real	45.
XLONLOC	real	0.
XXHATLOC	real	20000.
XYHATLOC	real	20000.
NILOC	integer	4
NJLOC	integer	4

- LGEOSBAL : Switch to fulfill the geostrophic balance or not
  - .TRUE. the geostrophic balance is satisfied by the initial fields
  - .FALSE. the geostrophic balance is not satisfied by the initial fields
- CFUNU : String of 3 characters, describing the type of function, which gives the x component of the wind. Possible configurations are listed below
  - 'ZZZ' : U = U(z). The U(z) values are taken from the Radio-Sounding or analitycal profile given in the free-formatted part of the PRE_IDEA1.nam file.
  - 'Y*Z' : U= F(Y)*U(Z). The U(z) values are build in the same way as the 'ZZZ' case and the function F(Y) is a simple function of Y, which must be adapted by modifying its definition directly in the routine FUNUY. The default function is :

$$FUNUY(\hat{y}) = \frac{1}{\cosh\left(\frac{\hat{y}-\hat{y_0}}{zwidth}\right)}$$

– 'Y,Z' : U= G(Y,Z). The function G must also be adapted by modifying its definition directly in the routine FUNUYZ. The default function is :

$$FUNUYZ(\hat{y},z) = \frac{1}{\cosh\left(\left(\frac{\hat{y}-\hat{y_0}}{zwidthy}\right)^2 + \left(\frac{z-z_0}{zwidthz}\right)^2\right)}$$

Notice that in this case the U(z) values given by the profile are not used.

- CFUNV : String of 3 characters, describing the type of function, which gives the y component of the wind. Possible configurations are listed below

  – 'ZZZ' : V = V(z). The V(z) values are taken from the Radio-Sounding or analitycal profile given in the free-formatted part of the PRE_IDEA1.nam file.

  – 'X*Z' : V= F(X)*V(Z). The V(z) values are build in the same way as the 'ZZZ' case and the function F(X) is a simple function of X, which must be adapted by modifying its definition directly in the routine FUNVX. The default function is :

$$FUNVX(\hat{x}) = \frac{1}{\cosh\left(\frac{\hat{x}-\hat{x_0}}{zwidth}\right)}$$

  – 'X,Z' : V= G(X,Z). The function G must also be adapted by modifying its definition directly in the routine FUNVXZ. The default function is :

$$FUNVXZ(\hat{x},z) = \frac{1}{\cosh\left(\left(\frac{\hat{x}-\hat{x_0}}{zwidthx}\right)^2 + \left(\frac{z-z_0}{zwidthz}\right)^2\right)}$$

  Notice that in this case the V(z) values given by the profile are not used.

- CTYPELOC : Type of informations used to give the localization of vertical profile (string of 6 characters)

  – 'IJGRID' for (i,j) point on index space

  – 'XYHATM' for (x,y) coordinates on conformal plane or cartesian plane

  – 'LATLON' for (latitude,longitude) on spherical earth

- XLATLOC : Latitude (in degrees) of the vertical profile localization (used in case CTYPELOC='LATLON')

- XLONLOC : Longitude (in degrees) of the vertical profile localization (used in case CTYPELOC='LATLON')

- XXHATLOC : position (in meters) x of the vertical profile localization (used in cases CTYPELOC='XYHATM')

- XYHATLOC : position (in meters) y of the vertical profile localization (used in cases CTYPELOC='XYHATM')
- NILOC : position i of the vertical profile localization (used in cases CTYPELOC='IJGRID') If you use a 1D model, then NILOC is reset to 2 by the program.
- NJLOC : position j of the vertical profile localization (used in cases CTYPELOC='IJGRID') If you use a 1D or a 2D model, then NJLOC is reset to 2 by the program.

3.2.18 Namelists of the externalized surface

The further definition of the surface parameters is not done by MESONH itself, but by the externalized surface included in it. So you are invited to refer to the **documentation of the surface**.

Two cases are encountered:

1. You do not have any input PGD file or you do not want to use the surface fields included in it (LREAD_GROUND_PARAM = .FALSE.). Then, you must both define the physiographic and prognostic fields, and you must fill the following namelists:
   - NAM_PGD_SCHEMES
   - NAM_COVER
   - NAM_ISBA (if you chose to use the ISBA scheme).
   - NAM_CH_EMIS_PGD
   - NAM_DUMMY_PGD
   - NAM_PREP_SURF_ATM
   - NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
   - NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
   - NAM_PREP_TEB (if you chose to use the TEB urban scheme)
   - NAM_PREP_ISBA (if you chose to use the ISBA scheme)
2. You do want to use all the informations contained in a PGD file. Then, only the prognostic variables must be defined, and the following namelists must be filled:
   - NAM_PREP_SURF_ATM
   - NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
   - NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
   - NAM_PREP_TEB (if you chose to use the TEB urban scheme)

- NAM_PREP_ISBA (if you chose to use the ISBA scheme)

Note that orography either comes from :

- the input PGD file (if any and if LREAD_ZS =.TRUE.). In this case, the atmospheric orography is also set equal to the one in this input PGD file.
- or from the orography you have defined from the MESONH namelists (in this case, the surface orography is forced to be equal to the atmosphere orography).

3.2.19 Example of namelist-part of PRE_IDEA1.nam :

```
&NAM_REAL_PGD  CPGD_FILE  = 'ALPES-CPGD'
               LREAD_ZS = .TRUE.,
               LREAD_GROUND_PARAM = .TRUE.  /
&NAM_CONF_PRE  LCARTESIAN=.TRUE.,
               LBOUSS=.FALSE.,
               CEQNSYS='DUR',
               CIDEAL='RSOU',
               CZS='FLAT',
               LPERTURB= .FALSE.,
               NVERB=5 /
&NAM_CONFn     LUSERV=.TRUE.,
               NSV_USER = 0 /
&NAM_DIMn_PRE  NIMAX=30,
               NJMAX=30 /
&NAM_VER_GRID  NKMAX=16,
               YZGRID_TYPE = 'MANUAL',
               ZDZGRD=70.  ZDZTOP=70.,
               ZZMAX_STRGRD=1000.    ,
               ZSTRGRD=0., ZSTRTOP= 0. /
&NAM_LUNITn    CINIFILE='ALP_ini' /
&NAM_DYNn_PRE  CPRESOPT ='RICHA',
               NITR=8, XRELAX=1.0 /
&NAM_LBCn_PRE  CLBCX(1)='CYCL', CLBCX(2)='CYCL',
               CLBCY(1)='CYCL', CLBCY(2)='CYCL' /
&NAM_VPROF_PRE CTYPELOC='IJGRID',
               NILOC=10, NJLOC=10,
```

```
                CFUNU='ZZZ', CFUNV='ZZZ',
                LGEOSBAL=.FALSE. /
&NAM_GRn_PRE    CSURF='EXTE'/
&NAM_CH_MNHCn_PRE LCH_INIT_FIELD = F /
&NAM_BLANK /
```

3.3 Free-format part

Each section of the free format part must be introduced by its corresponding keyword (writen on a separated line)

There is always a moist variable written in PRE_IDEA1.nam file, even in idealized dry cases, for which the moist variable should be equal to zero in the PRE_IDEA1.nam file. The produced initial file will always contain a moist variable in 'CSTN' and 'RSOU' cases.

3.3.1 Optional Vertical grid :

keyword: **ZHAT**

If the vertical grid generation selector CZGRID_TYPE is equal to 'MANUAL', then you must enter just under the namelist part, the heights of the vertical velocity levels. You must start from the ground level (K=2) to the model top (K=KMAX +2), thus you only have to enter KMAX + 1 values. For instance, for the PRE_IDEA1.nam described above, it leads to 17 levels.

3.3.2 Radiosounding case :

keyword: **RSOU**

The radiosounding data are written in the free-format part of PRE_IDEA1.nam file, where the altitude variable is :

- the pressure in case KIND='STANDARD' or '**P**UVTHVMR' or '**P**UVTHVHU' or '**P**UVTHDHU' or '**P**UVTHDMR' (real, in Pascal)
- the height in case '**Z**UVTHVMR' or '**Z**UVTHVHU'or '**Z**UVTHDMR' or '**Z**UVTHLMR' (real, in meters)

The first wind variable is :

- the wind direction in case KIND='STANDARD' (real,in degrees)

- the zonal wind in cases KIND='P**UV**THVMR' or 'P**UV**THDMR' or 'Z**UV**THDMR' or 'Z**UV**THLMR'or 'Z**UV**THVHU' or 'P**UV**THDHU' or 'Z**UV**THVMR' or 'P**UV**THVHU' (real, in m/s)

The second wind variable is :

- the force direction in case KIND='STANDARD' (real, in m/s)
- the meridian wind in cases KIND='P**UV**THVMR' or 'P**UV**THDMR' or 'Z**UV**THDMR' or 'Z**UV**THLMR' or 'Z**UV**THVHU' or 'P**UV**THDHU' or 'Z**UV**THVMR' or 'P**UV**THVHU' (real, in m/s)

The temperature variable is :

- the temperature in case KIND='STANDARD' (real, in Kelvin)
- the virtual potential temperature in cases KIND='PUV**THV**MR' or 'PUV**THV**HU' or 'ZUV**THV**MR' or 'ZUV**THV**HU' (real,in Kelvin)
- the dry potential temperature in cases KIND='PUV**THD**MR' or 'PUV**THD**HU' or 'ZUV**THD**MR'(real, in Kelvin)
- the liquid potential temperature in case KIND='ZUV**THL**MR'(real, in Kelvin)

The moist variable is :

- the dew point temperature in case KIND='STANDARD' (real, in Kelvin)
- the vapor mixing ratio in cases KIND='PUVTHV**MR**' or 'ZUVTHD**MR**' or 'ZUVTHV**MR**' or 'PUVTHD**MR**' (real, in Kg/Kg)
- the total water mixing ratio in cases KIND= 'ZUVTHLMR (real, in Kg/Kg)
- the relative humidity in cases KIND= 'ZUVTHV**HU**', or 'PUVTHD**HU**' or 'PUVTHV**HU**' (real, in percents)

Additional cloud variables

For the moment, this configuration works only for KIND='PUVTHDMR' or 'ZUVTHDMR' and L1D=.TRUE.. It is planned to compute radiation diagnostics with the **DIAG** program (see chapter 6).

- cloud mixing ratio if LUSERC=T or LUSERI=T (real, in Kg/Kg)
- ice mixing ratio if LUSERI=T (real, in Kg/Kg)

You should take care that the levels are dense enough so that the Laplace relation, which gives the thickness between successive levels, can be applied. The radiosounding informations are written in the file in the following order :

- YEAR (integer, exemple : 1994), MONTH (integer, exemple : 4), DAY (integer, exemple : 22), TIME (real, in seconds, exemple : 36000 for 10 h)
- KIND of data used for the radiosounding (string of 8 charcaters) Nine kind of data are possible : 'STANDARD', 'PUVTHVMR', 'PUVTHVHU', 'ZUVTHVMR', 'ZUVTHVHU', 'PUVTHDMR', 'PUVTHDHU', 'ZUVTHDMR', 'ZUVTHLMR'.

  Except for the STANDARD kind :

  - the first letter of KIND represents the kind of altitude variable (P for pressure and Z for height),
  - the second and third letters represent the kind of wind variables (U for zonal wind, V for meridian wind),
  - the fourth, fifth and sixth letters represent the kind of temperature variable (THV for virtual potential temperature, THD for dry potential temperature and THL for liquid potential temperature),
  - the seventh and heighth letters represent the kind of moist variable (HU for relative humidity and MR for vapor mixing ratio).

  (In case of STANDARD kind, the altitude variable is the pressure, the wind variables are direction and force of wind, the temperature variable is the temperature and the moist variable is the due point temperature.)
- HEIGHT of GROUND LEVEL (real, in meters)
- PRESSURE at GROUND LEVEL (real,in Pascal)
- a TEMPERATURE variable at GROUND LEVEL (real, in Kelvin)
- a MOIST variable at GROUND LEVEL
- NUMBER WIND data LEVELS (integer)
- level 1 : ALTITUDE variable , first WIND variable, second WIND variable at wind level 1 (the lowest wind-level).
- level 2 : ALTITUDE variable, first WIND variable, second WIND variable.

  ⋮

  ⋮

- uppermost wind level : ALTITUDE variable, first WIND variable, second WIND variable.
- NUMBER of mass data LEVELS (integer) **Note that this number includes the ground level (i.e. the first level).** That is why the following list starts at level 2.
- level 2 : ALTITUDE variable, TEMPERATURE variable, MOIST variable, additional cloud variable(s) (the mass level 1 is at ground).
- level 3 : ALTITUDE variable, TEMPERATURE variable, MOIST variable, additional cloud variable(s) .

⋮

⋮

- uppermost mass level: ALTITUDE variable, TEMPERATURE variable, MOIST variable, additional cloud variable(s)

Example of free part of PRE_IDEA1.nam

```
RSOU
1990 10 3 72000.
'STANDARD'
200.
100240.
287.5
276.
2
85000. 20. 10.
70000. 30. 10.
3
90000. 280. 275.
60000. 271. 269.
```

3.3.3 Constant moist Brunt-Vaisala case :

keyword: **CSTN**

Data of the vertical profile are written in the free-format part of PRE_IDEA1.nam file in the following order :

- YEAR (integer, example : 1994), MONTH (integer, example : 4), DAY (integer, example : 22), TIME (real, in seconds, example : 36000. for 10 h)
- NUMBER of LEVELS (integer)
- VIRTUAL POTENTIAL TEMPERATURE at GROUND LEVEL (*i.e* at the first level) (real,in Kelvin)
- PRESSURE at GROUND LEVEL (*i.e* at the first level) (real, in Pascal)
- HEIGHT at all levels. **the first level is the ground level**
- ZONAL WIND COMPONENT at all levels (the first level is the ground level)
- MERIDIAN WIND COMPONENT at all levels (the first level is the ground level)
- RELATIVE HUMIDITY at all levels (the first level is the ground level)
- MOIST BRUNT VAISALA FREQUENCY at all layers (the number of layers is the number of levels - 1)

In this case, the level number can even be equal to 1, because the profile informations are linearly interpolated on the model grid without orography (wind components, θ_v and humidity) before the application of the Laplace relation to deduce the pressure and the vapor mixing ratio. Thus, the layers' thicknesses are never too large to invalidate the Laplace relation.

Example of free part of PRE_IDEA1.nam

```
CSTN
2006 06 06 21600.
5
287.5
100240.
200. 1000. 1500. 3000. 4000.
10. 20. 25. 30. 35.
2. 10. 12.5 11.5 15.
80. 84. 85. 79. 87.
0.01 0.014 0.015 0.016
```

3.3.4 The forced version

keyword: **ZFRC** or **PFRC**

For idealized simulations a forced mode can be useful to impose the effects of a simplified large scale environment to the model solution. This functionality works (LFORCING=.TRUE. in module MODD_CONF) when CIDEAL='RSOU' or 'CSTN' (see 5.2.10 and 5.3) and only in the case LCARTESIAN =.TRUE. and LGEOSBAL =.FALSE. for inclusion of a geostrophic wind forcing. All forcing fields issue from spatial interpolation of chronological series of 1D data provided by the user onto the model grid. They are prepared during the **prep_ideal_case** sequence and are stored in the LFI files for further use in case of RESTART model run.

The forcing fields can be time dependent. Application of the forcing begins as soon as the date and time of the first set of forcing field given by the user, is lower or equal to the current date and time of the model run. The forcing action of the last forcing field is remanant, this is a way to impose a stationnary forcing. When the current date and time of the model run is bounded by two successive forcing fields, a simple linear interpolation in time is made.

Note that an available Newtonian relaxation forcing type on $[u, v]$ and/or $[\theta, r_v]$ is exclusive from the other physical forcings.

The forcing informations and soundings have to be added at the end of the free-format part already written for CIDEAL='CSTN' or 'RSOU'. First, the type of forcing and the number of time dependent forcing are given:

- keyword forcing type (character*4)
  - ZFRC means that the altitude of the forcing data are in height scale (meters).
  - PFRC means that the altitude of the forcing data are in pressure scale (Pascal).
- number of time dependent forcing (integer)

The 1D forcing data are different from the one used to initialize the model because specific data have to be entered. The data used to define each forcing are given sequentially in the following order (one item per line):

- date and time of the forcing in the format:

  year (integer),

  month (integer),

  day (integer) and

  time of the day (real, s).
- ground height (real, m)
- ground pressure (real, Pa) (ATTENTION: in the MASDEV3_1 version either the ground height or the surface pressure was read, now we read both !)

- θ_d (real, K) at ground level (Nota: it is used later in the code to compute - if asked - a time varying sea surface temperature).
- r_v (real, kg/kg) at ground level
- number of level (integer)
- height of level1 (real, m) if ZFRC or pressure at level1 (real, Pa) if PFRC,

  u_{frc} component at level1 (real, m/s),

  v_{frc} component at level1 (real, m/s),

  θ_{frc} at level1 (real, K),

  $r_{v\ frc}$ at level1 (real, kg/kg),

  w_{frc} at level1 (real, m/s),

  $(\partial\theta/\partial t)_{frc}$ at level1 (real, K/s) and

  $(\partial r_v/\partial t)_{frc}$ at level1 (real, 1/s).

  $(\partial\theta/\partial x)_{frc}$ at level1 (real, K/m).

  $(\partial\theta/\partial y)_{frc}$ at level1 (real, K/m).
- idem at level2
- ...
- idem at levelN

If PFRC is the forcing type, an additional sounding is given in order to convert the pressure levels into height levels with enough accuracy. Data are organized as follows:

- number of level (integer)
- pressure at level1 (real, Pa),

  θ at level1 (real, K) and

  r_v at level1 (real, kg/kg).

  This operation is repeated until the previous number of sounding is reached.

  Example of free part of PRE_IDEA1.nam

```
ZFRC
   1
1983 07 01 0.
```

```
0
284.5
 .008
 6
   5.   -7.0   0.0  281.10   0.00540     -0.00000   0.0  0.0
  15.   -7.0   0.0  281.10   0.00540     -0.00000   0.0  0.0
1095.   -7.0   0.0  280.75   0.00540     -0.00300   0.0  0.0
1145.   -7.0   0.0  290.60   0.00190     -0.00300   0.0  0.0
3000.   -7.0   0.0  304.15   0.00190     -0.00300   0.0  0.0
9000.   -7.0   0.0  346.15   0.00190     -0.00300   0.0  0.0
```

3.3.5 Discretized orography

keyword: **ZSDATA**

Only the orography corresponding to the computational domain must be provided in free format. For 3D orography, data are read like if it was a map (the first line is the Northern border and the first data is the North-West corner) with one line per Y-axis increment.

Example of free part of PRE_IDEA1.nam

```
ZSDATA
30.    30.    35.     50.     30.     30.
30.    59.5  133.3   100.2   136.7   100.
35.    89.5  183.3   200.2   299.7   170.5
50.    112.5 193.0   210.2   206.7   120.
40.    82.5  153.0   180.5   156.7   100.3
```

3.4 The output MESONH-file

The descriptive part (file DESFM) of the MESONH-file is built with the default values associated with all the degrees of the model simulations and with some informations present in the PRE_IDEA1.nam file (name of the file, activations of the different moist variables...). Therefore, most of the informations present in the descriptive part DESFM (for instance type of the turbulence, amplitude of the Rayleigh damping...) are not really related to the description of the generation of the LFI file. This is due to the way of building this initial file not from a MESONH simulation but from external informations (given in PRE_IDEA1.nam). A better way of filling this descriptive part, would have been to generate empty namelists for this kind of informations in the LFI file but in Fortran 90, it is still impossible to write empty namelists, they have to include at least one element, we therefore decide to write complete namelists, filled by the default parameters when these variables were not initialized.

In conclusion, not all the informations present in the descriptive part of the initial MESONH file are significant. The specification of the segment to perform with this MESONH file (used as starting point) will be realized in the EXSEG1.nam file, described in chapter 4.

3.5 An illustrative example of PREP_IDEAL_CASE:

This section presents a complete example of a preparation of initial conditions for an academic study: it explains how to build on the "local_host" the UNIX job, which will execute the fortran program on the "remote_host".

The selected case is the following:

- 2D mountain
- moist one layer atmosphere

The first step is the control of the initialization for the control variables. Enter the following commands on the "local_host", when you are logged in:

```
echo $SIMUL
```

this is the working directory. If it is not the right name, change it by:

```
SIMUL=myworkdir ; export SIMUL
```

now, go in this directory:

```
cd $SIMUL
```

In one configuration file (`.profile, .bash_profile, ...`) set the other environmental variables:

- PATH must contain $MESONH/procedures or enter the following command:

  ```
  PATH=$PATH:$MESONH/UPDATE_PROCEDURES:$MESONH/procedures
  ```

- MANPATH must contain `$MESONH/procedures` or change it as for PATH
- EDITOR=/usr/bin/vi

You are now ready to start to work. Copy in this directory all the files present in $MESONH/procedures ending by `rc` in $SIMUL. In this example, we will need to modify `prepmodelrc, tosuprc` with any editor, to obtain the following files:

FILE prepmodelrc

```
#! /bin/sh
#
# default input variables for prepmodel
#
#  N. Asencio 21/09/94
#
# NEW ! Improvements in prepmodelrc are added at the beginning of the file, read them !
#-------------------------------------------------------------------------------
#
                    #                comfort variables
#OUTSCRIPT=         # instead of outprepmodel, outprepideal, outprep_pgd,
                    # outprepreal, outprepspawn, outprepdiag
#SLEEPSECONDS=      # instead of 10
EDITOR=vi
#
#
# specific variables
#
#      Control tools commands (in prepmodel global var) yes or no (default variable)
TOOLSCONTROL=no
#      tori hpce tora (fuji until next monday) lx____ (32bit)
#      a must have
OUTDEST=
#
#  Automatic submit by tosupc for part one, tosupcrc with TIME MEM NBP TPN has to be OK
LSOUMISAUTO=T
                    # options for treatment of all FM files (and post-treatment)
                    # 'lfiz' 'lfiz conv2dia' 'fm unlfiz' 'all' 'fmmore'
OUTFILE_TOOLS=''
                    # SUBMIT_NEXTJOBS = 0 or filenames to submit super_calculator
                    # at the end of this job
                    # tosupcrc must be present on the same directory
                    # 0=no job
                    # one or several scripts between commas, which will
                    # be submitted one by one by tosupc
                    # the pathnames are relative to the current $SIMUL directory
                    # or absolute
SUBMIT_NEXTJOBS=0
                    # LOCAL_NEXTJOBS = 0 or filenames to execute workstation jobs
                    # at the end of this job
                    # list of jobs executed on local workstation at the end of this job;
                    # second argument = list of <sh command arguments> which will run
                    # on your workstation with "at" procedure
                    # the pathnames are relative to the current $SIMUL directory or absolute
                    # 0=no job
                    # LOCAL_NEXTJOBS = "dirlocal trace1"
                    # the workstation job will execute : cd dirlocal ; sh trace1
LOCAL_NEXTJOBS=0
                    # name of your personal makefile
                    # if you don't have one, general makefile will be executed
                    # on the remote machine, according its OS
MKFNAME=make_mnh
                    # user binary library (0 if none, will be searched from your $HOME)
BIBUSER=0
                    # reference binary library
BIBMASTER=$DEFBIBMASTER
                    # reference bugfix binary library
BIBBUGFIX=$DEFBIBBUGFIX
                    # run with debug options or not (run/debug)
```

```
DEBUG=run
                  # loading options added to the default ones :-M|-f|-l perf...
LOAD_OPT=''
        ################ NEW PROGRAM ZOOM_PGD since masdev4_7 ##################
        # are also available PREP_IDEAL_CASE/MODEL/PREP_PGD/DIAG/SPAWNING/PREP_REAL_CASE
        #                    PREP_NEST_PGD
        # other names are possible, in this case do not forget to fill in NAMELISTFILE=
MAINPROG=0
                  # special filename for input namelists
                  # for ZOOM_PGD default namelist is PRE_ZOOM
NAMELISTFILE=default
                  # special list of variables getting from $NAMELISTFILE
LISTGET=default
                  # ---- input files ---------
                  # location of the input FM files on execution machine
                  # or storage machine ($HOME/$$INDIR) -supc/archiv
INHOST=supc
                  # to get the input FM files, directory name starting
                  # at $HOME if it begins with home/
                  # at $workdir if it begins with work/
                  # indicate one or several directories between double quotes
INDIR="work/INOUTFILES"
                  # user name on which the get will be executed
                  # 0=$LOGNAME
INLOGIN=0
                  # Overview of archiv case:
#INHOST=archiv
#INDIR="../../mxxx/mxxx007/CHEMIN ../mgrp999/NEXTDOOR"
#INLOGIN=0
                  # OR
#INHOST=archiv
#INDIR=NEXTDOOR
#INLOGIN=mrgp999
                  # ---- output files ---------
                  # storage of the output FM files on execution machine
                  # or storage machine -supc/archiv/$RMACH/aerosv2......
OUTHOST=supc
                  # to put the output FM files directory name starting
                  # at $HOME if it begins with home/
                  # at $workdir if it begins with work/
OUTDIR=work/INOUTFILES
#
# global variables used in prepmodel
#
#DEBUGSCRIPT=                            ;# ON /OFF
#ENVIRONMENT=                            ;# SILENTINTERACTIVE/INTERACTIVE
```

FILE tosupcrc

```
#! /bin/sh
#
# default input variables for tosupc (default values for the NQS parameters)
#
#  N. Asencio 13/12/94
#-------------------------------------------------------------------------
#
# specific variables
#
                # filename which contains the script to be run on remote-host
```

```
JOBFILE=${1:-0}
                  # jobname
JOBNAME=${2:-$(basename $JOBFILE)}

################### tosupc will not work unless you correctly fill TIME MEM TPN NBP ####
                  # time in seconds for J2&J3 jobs
TIME=
                  # memory ex: =2000 =2Gb =128Mb (tori: less than 128Gb per node)
                  # be careful : reduce the memory for multi-tasks jobs
MEM=
                  ### tasks per node: only for prep_ideal_case, run or diag
                  ### number of used CPUs from 1 to 8
  ### The most important for parallel execution!
TPN=1
                  # mono or multi-node on NEC (1 to 4) and on IBM
NBP=1
################### tosupc will not work unless you correctly fill TIME MEM TPN NBP ####

                  # Sending a mail abort + beg and/or at the end job (begend/beg/end/no)
                  # default is abort, =no is no mail at all
MAIL=

#
# global variables used in tosupc
#
#DEBUGSCRIPT=                           ; #ON /OFF
#ENVIRONMENT=SILENTINTERACTIVE          ; #SILENTINTERACTIVE/BATCH/INTERACTIVE
```

All the necessary informations for the UNIX to be correctly executed on the remote_host will be read during this preparative job on the different `rc` files on the local_host.

Now, you may create a file which contains the following namelists:

FILE PRE_IDEA1.nam

```
&NAM_DIMn_PRE  NIMAX=128, NJMAX=1 /
&NAM_VER_GRID  NKMAX=32, YZGRID_TYPE = 'FUNCTN', ZDZGRD=500., ZDZTOP=500.,
               ZZMAX_STRGRD=1000.    , ZSTRGRD=0., ZSTRTOP= 0.,
&NAM_CONFn     LUSERV=.TRUE., NSV_USER = 0 /
&NAM_GRID_PRE  XLAT0 = 48.25 , XLON0 = 0.,
               XRPK  = 0.    , XBETA = 0.,
               XLONORI = 48.25, XLATORI = 0. /
&NAM_CONF_PRE  LCARTESIAN=.TRUE., LBOUSS=.FALSE.,
               CIDEAL='CSTN', CZS='BELL',
               LPERTURB= .FALSE., NVERB=1 /
&NAM_GRIDH_PRE XDELTAX=5.E2 , XDELTAY=5.E2,
               XHMAX=500., XAX=10.E3, XAY=10.E3, NIZS=64, NJZS=2,
               NEXPX = 1, NEXPY=1 /
&NAM_LUNITn    CINIFILE='HYD2D' /
&NAM_DYNn_PRE  CPRESOPT ='RICHA', NITR=4, XRELAX=1.0 /
```

```
&NAM_LBCn_PRE  CLBCX(1)='OPEN', CLBCX(2)='OPEN',
               CLBCY(1)='OPEN', CLBCY(2)='OPEN' /
&NAM_VPROF_PRE CTYPELOC='IJGRID', NILOC=10, NJLOC=2,
               CFUNU='ZZZ', CFUNV='ZZZ',
               LGEOSBAL=.FALSE. /
&NAM_GRn_PRE   CSURF='EXTE' /
&NAM_CH_MNHCn_PRE LUSECHEM = F /
CSTN
2
285.
100000.
0. 20000.
10. 10.
0. 0.
40. 40.
0.01
```

This file contains the necessary informations necessary to generate the initial conditions for a quasi-hydrostatic flow, in the weakly non-linear regime, with a regular vertical grid.

Now, you run the preparative script by :

`prepmodel`

and you have to answer to the questions, asked by the script:

`please enter a name for your simulation directory`

answer: . (if `SIMUL` is not initialized)

Another way to prepare the file `outprepideal` on local_host, is to set ENVIRONMENT to interactive:

`ENVIRONMENT=INTERACTIVE ; export ENVIRONMENT`

The different parameters, set to the values written in the `rc` files will then appear at the screen and can be modified with your keyboard. This configuration is only available on CNRM workstations.

Then you submit to the remote_host for execution the output file `outprepideal` by:

`tosupc outprepideal`

In this case, the final file `outprepideal` will be send in the batch queues of the remote_host and the execution listing report will come back in the local_host, the FM-file generated during

this step will be disposed in the directory, mentioned in `prepmodelrc` (`OUTDIR`). If at contrary, you do not submit the preparative job by `tosupc`, you can login on the remote_host, get this file from the local_host by ftp and execute it in an interactive session on the remote_host (in $tmpdir directory) by the following command:

```
outprepideal
```

In this case, the only difference is that the report is directly printed at the screen. Please, note that the number of points is very weak in this example and therefore, you do not have problems with the memory allocation, but for more points you can overshoot the maximum memory available in an interactive session and in this situation, you must dispose your `outprepideal` file in the batch queues of the remote_host, to obtain more available memory.

Chapter 4

Perform a MESONH simulation:

4.1 What does go in and out?

The MESONH simulation model used previously prepared MESONH files, to initialize the prognostic variables of the model and all the other informations necessary to perform the simulation (like the computational grid for instance). It can also need additional files to realize the coupling of the outermost model with a large-scale model.

The MESONH user will specify some free parameters of the run by fixing their new values in the NAMELISTs of the file EXSEG$n.nam.

When more than one model is present, each model needs its own MESONH file to be initialized and its own EXSEG$n.nam file to fix the free-parameters (note that a lot of physical free-parameters depends on the mesh and therefore vary with the model number).

The input files are read by the program in order to realize the initialization and the eventual coupling of the MESONH model with a large-scale model (CEP, Arpège...). They all have the same standart form, described in the first Annexe:

- a descriptive part. It is filled by the copy of the EXSEG$n.nam file corresponding to the model number in the grid nesting procedure, which has been used to perform this segment. And at the end of these informations are added the comments describing the fields written in the binary part of the file.
- a binary part, where the fields are written by the LFI subroutines

The output files are of two types:

- synchronous files for a given instant of the run. They contain the prognostic fields and eventually, additionnal records for supplementary diagnostic fields at the same instant. The file name ends by `00n` with `n`>0
- a diachronic file for the temporal series of prognostic or diagnostic fields. The file name ends by `000`

4.2 The input EXSEG$n.nam file

We now describe in the following subsection the different NAMELISTs present in a complete EXSEG$n.nam file. Each variable present in a Namelist of the EXSEG$n.nam file belongs to a declarative module whose name is related to the NAMelist name:

NAM_xxxxx $\Longrightarrow$ MODD_xxxxx

The documentation of the declarative modules MODD_xxxxx is present in the Fortran code and contains a description of each variable present in the Namelist NAM_xxxxx. Thus, we only give the list of the subset of MODD_xxxxx present in the Namelist NAM_xxxxx with a short description of each parameter.

In EXSEG$n.nam file, the order of namelists is free, not all the free-parameters need to be fixed then unset namelists can be ommited. For instance, if no value is present for the variable CPRESOPT in the NAMELIST NAM_DYNn of EXSEG2.nam (the index 2 is for model 2), the model will take the value present in the MESONH file, used to initialize the model 2 for this segment. This information is present in the descriptive part of the MESONH file (see Chapter 3). If it is also absent from the MESONH initial file, the model will use the default value written in the Fortran code of the model.

We now list the namelists:

- Firstly, namelists common to all the nested models (without suffix n) are presented **alphabetically**, from 4.2.1.
- Secondly namelists relative to one model (with suffix n), from 4.2.11.
- Thirdly namelists relative to the surface model, from 4.2.21.
- Fourthly, namelists relative to chemistry, from 4.2.22.
- Fifthly, namelists relative to on-line diagnostics (balloons, aircrafts, profilers, stations, series), from 4.2.25.
- And endly namelists relative to budgets, from 4.2.29, and LES budgets, from 4.2.43.

4.2.1 Namelist NAM_BLANK (available variables)

Fortran name	Fortran type	default value
XDUMMY1 .. XDUMMY8	real	0.
NDUMMY1 .. NDUMMY8	integer	0
LDUMMY1 .. LDUMMY8	logical	TRUE
CDUMMY1 .. CDUMMY8	80 characters	”
XDUMMY	array(real)	20* 0.
NDUMMY	array(integer)	20* 0
LDUMMY	array(logical)	20* TRUE
CDUMMY	array(80 characters)	20* ”

Eight dummy, real, integer, logical and character*80 variables and arrays of dummy, real, integer, logical and character*80 for test and debugging purposes are defined and passed through the namelist read operations. None of the MesoNH routines uses any of those variables. When a developper choses to introduce temporarily a parameter to some subroutine, he has to introduce a USE MODD_BLANK statement into that subroutine. Then he can use any of the variables defined here and change them easily via the namelist input.

4.2.2 Namelist NAM_CONF (global configuration parameters)

It contains the model configuration parameters common to all the models. They are included in the module MODD_CONF.

Fortran name	Fortran type	default value
CCONF	5 characters	’START’
LFLAT	logical	FALSE
CEQNSYS	3 characters	’DUR’
LFORCING	logical	FALSE
NMODEL	integer	1
NVERB	integer	5
NHALO	integer	1
CSPLIT	10 characters	’YSPLITTING’
LLG	logical	FALSE
LINIT_LG	logical	TRUE
CINIT_LG	5 characters	’FMOUT’
LNOMIXLG	logical	FALSE
CEXP	5 characters	’EXP01’
CSEG	5 characters	’SEG01’

- CCONF: configuration of all models
  - ’START ’ for start configuration
  - ’RESTA’ for restart configuration
- CEQNSYS: Equation system resolved by the MESONH model

  - 'LHE' Lipps and HEmler anelastic system
  - 'DUR' approximated form of the DURran version of the anelastic sytem
  - 'MAE' classical Modified Anelastic Equations but with not any approximation in the momentum equation

- LFLAT: Switch for zero ororography
  - .TRUE. = no orography (zs=0.)
  - .FALSE. = the orography is not zero everywhere
- LFORCING: Switch to use forcing sources
  - .TRUE. add forcing sources
  - .FALSE. no forcing sources
- NMODEL: Number of nested models
- NVERB: Level of informations on output-listing
  - 0 for minimum of prints
  - 5 for intermediate level of prints
  - 10 for maximum of prints
- NHALO: Size of the halo for parallel distribution. This variable is related to computer performance but has no impact on simulation results
- CSPLIT: Kind of domain splitting for parallel distribution. This variable is related to computer performance but has no impact on simulation results
  - 'BSPLITTING' domain is decomposed in Box along X and Y
  - 'XSPLITTING' the X direction is splitted in stripes along Y
  - 'YSPLITTING' the Y direction is splitted in stripes along X
- LLG: Switch to use LaGrangian variables
- LINIT_LG: Switch to reinitialize LaGrangian variables
- CINIT_LG: When reinitialize LaGrangian variables :
  - 'FMOUT' each time an output file is written

– other string: only when starting a new segment (CCONF='RESTA')

- LNOMIXLG: Switch to not use turbulence for LG variables
- CEXP: Experiment name (this is the name of the set of run, you have performed or you want to perform on the same physical subject) **Please do not leave any blank character in this name!**
- CSEG: Name of segment (this is the name of the future run, you want to perform) **Please do not leave any blank character in this name!**

From these two last informations, we built the names of the different MESONH output files:

CEXP.$n.CSEG.nbr

where $n represents the number of the model which generates this output and nbr is the number of the outfile. For instance, if $CEXP = HYDRO'$ and $CSEG =' INIT1'$ and we use only one model (no gridnesting) the different output will be called:

$HYDRO.1.INIT1.001$, $HYDRO.1.INIT1.002$, ...

4.2.3 Namelist NAM_DUST

This namelist is use to activate explicit aerosols dusts. It is not necessary to use chemistry to activate dusts but it is recommended to activate on-line dust emissions (see surface namelists). Radiative direct effects are automatically deduced from an interpolation table of SHDOM radiative code (Mie).

Fortran name	Fortran type	default value
LDUST	logical	FALSE
LVARSIG	logical	FALSE
LSEDIMDUST	logical	FALSE
NMODE_DST	integer	3

- LDUST: switch to activate passive dust aerosol.
- LVARSIG: switch to activate variable standard deviation for each dust modes.
- LSEDIMDUST: switch to activate dust sedimentation.
- NMODE_DST: number of lognormal dust modes (maximum of 3 modes).

4.2.4 Namelist NAM_DYN (global parameters for the dynamics)

Fortran name	Fortran type	default value
XSEGLEN	real	43200.
XASSELIN	real	0.2
XASSELIN_SV	real	0.02
LCORIO	logical	TRUE
LNUMDIFF	logical	FALSE
LZDIFFU	logical	FALSE
XALKTOP	real	0.01
XALZBOT	real	4000.

It contains the dynamics parameters common to all models. They are included in the module MODD_DYN.

- XSEGLEN : Segment Length in seconds, corresponding to the duration of the segment simulation.
- XASSELIN : Amplitude of the Asselin temporal filter for meteorological variables
- XASSELIN_SV : Amplitude of the Asselin temporal filter for scalar variables
- LCORIO : Switch to set the Coriolis parameters f and f^* to zero
  - .TRUE. the Earth rotation is taken into account
  - .FALSE. the Earth rotation effects are neglected
- LNUMDIFF : Switch to activate the numerical diffusion (XT4DIFF in NAM_DYNn defines the intensity of this diffusion).
- LZDIFFU: Switch to apply the horizontal diffusion according to Zangl (2002) adapted to mountainous topography. No amplitude is applied for this type of diffusion (XT4DIFF in NAM_DYNn is inactive).
  - .TRUE. This horizontal diffusion is applied
  - .FALSE. This horizontal diffusion is not applied
- XALKTOP : Maximum value of the Rayleigh damping (in s^{-1})at the top of the upper absorbing layer. The shape of the absorbing layer is a $\sin^2$ of $\hat{z}$ (see the scientific documentation for more details).
- XALZBOT : Height (in meters) in the physical space of the upper absorbing layer base.

4.2.5 Namelist NAM_FMOUT (output instants)

Fortran name	Fortran type	default value
XFMOUT	array (real)	8*192* 999.

- XFMOUT(m,i) :is an array of increments in seconds from the beginning of the segment to the instant where the i-th fields output on FM-files is realized by model "m"

4.2.6 Namelist NAM_FRC (forcing control)

Application of a specific forcing is enabled by a dedicated switch. When a Newtonian relaxation is requested, the damping time XRELAX_TIME_FRC and the height (fixed or physically based) above which the forcing is applied, XRELAX_HEIGHT_FRC and CRELAX_HEIGHT_TYPE, must be set.

Fortran name	Fortran type	default value
LGEOST_UV_FRC	logical	FALSE
LGEOST_TH_FRC	logical	FALSE
LTEND_THRV_FRC	logical	FALSE
LVERT_MOTION_FRC	logical	FALSE
LRELAX_THRV_FRC	logical	FALSE
LRELAX_UV_FRC	logical	FALSE
XRELAX_TIME_FRC	real	10800.
XRELAX_HEIGHT_FRC	real	0.
CRELAX_HEIGHT_TYPE	character*4	'FIXE'
LTRANS	logical	FALSE
XUTRANS	real	0.
XVTRANS	real	0.
LPGROUND_FRC	logical	FALSE

- LGEOST_UV_FRC : switch to use a prescribed geostrophic wind.
  - .TRUE. to integrate a geostrophic wind with a constant Coriolis parameter $f = 2 \times \Omega \times$SIN(XLAT0). The LCORIO flag of module MODD_CONF must be .TRUE.
  - .FALSE. not active

- LGEOST_TH_FRC : switch to apply a large scale horizontal advection on the potential temperature field. The gradients result from the thermal wind balance.
  - .TRUE. to integrate an horizontal advection of θ.
  - .FALSE. not active

- LTEND_THRV_FRC : switch to simulate a large scale θ and humidity tendency.
  - .TRUE. to integrate a tendency for θ and r_v.

  - .FALSE. not active

- LVERT_MOTION_FRC : switch to simulate a large scale vertical transport of all the prognostic fields.
  - .TRUE. to integrate a vertical transport with an upstream scheme.
  - .FALSE. not active

- LRELAX_THRV_FRC : switch to apply a Newtonian relaxation on the potential temperature and humidity fields.
  - .TRUE. to relax θ and r_v towards large scale values.
  - .FALSE. not active

- LRELAX_UV_FRC : switch to apply a Newtonian relaxation on each horizontal wind component.
  - .TRUE. to relax the horizontal wind towards large scale values.
  - .FALSE. not active

- XRELAX_TIME_FRC : constant damping time for the forced relaxation.

- XRELAX_HEIGHT_FRC : height above which a forced relaxation is enabled when CRELAX_HEIGHT_TYPE='FIXE' or minimal height if 'THGR' is taken.

- CRELAX_HEIGHT_TYPE : definition of the height above which a forced relaxation is enabled.
  - 'FIXE' means that a forced relaxation is never applied below XRELAX_HEIGHT_FRC.
  - 'THGR' means that a forced relaxation is never applied below the maximal height between XRELAX_HEIGHT_FRC and the height above which $\partial\theta/\partial z$ is the highest for each column.

- LTRANS : switch to apply a Galilean translation of the domain of simulation
  - .TRUE. The translation speed of the domain of simulation will be XUTRANS,XVTRANS
  - .FALSE. : not active

- LPGROUND_FRC : switch to simulate a time varying ground pressure. (WARNING: THIS FLAG CAN NOT YET BE ACTIVATED !)

- .TRUE. means that the ground pressure is updated if it varies in time in the "Free Format Forcing Part" of the PRE_IDEA1.nam file.
- .FALSE. not active

4.2.7 Namelist NAM_NESTING (grid nesting configuration)

Fortran name	Fortran type	default value
NDAD	array (8 real)	m-1
NDTRATIO	array (8 integer)	1
XWAY	array (8 real)	2

- NDAD(m) : is the model number of the father of each model "m"
- NDTRATIO(m) : is the ratio between time step of model m and its father NDAD(m)
- XWAY(m) : is the interactive nesting level for model m and its father NDAD(m)
  - 1 one_way interaction
  - 2 two_way interaction
  - 3 two_way interaction : upward information to the father also for 2D fields (Surface precipitation and SW radiative fluxes) that are used by the surface.

4.2.8 Namelist NAM_PARAM_C2R2 (control variable of the 2-moment warm microphysical schemes C2R2 and KHKO)

Fortran name	Fortran type	default value
HPARAM_CCN	character (LEN=3)	'XXX'
HINI_CCN	character (LEN=3)	'XXX'
HTYPE_CCN	character (LEN=1)	'X'
XCHEN	real	0.0
XKHEN	real	0.0
XMUHEN	real	0.0
XBETAHEN	real	0.0
XCONC_CCN	real	0.0
XR_MEAN_CCN	real	0.0
XLOGSIG_CCN	real	0.0
XFSOLUB_CCN	real	1.0
XACTEMP_CCN	real	280.0
XALPHAC	real	3.0
XNUC	real	1.0
XALPHAR	real	1.0
XNUR	real	2.0
LRAIN	boolean	TRUE
LSEDC	boolean	TRUE
LACTIT	boolean	FALSE

It contains the control parameters used when calling the C2R2 warm microphysical scheme. They are in the declarative module MODD_PARAM_C2R2.

- HPARAM_CCN: Acronym of the CCN activation parameterization to use ('CPB', 'TFH' or 'TWO'). The 'TFH' and 'TWO' need only to prescribe the XCHEN and XKHEN parameters.
  - 'TWO' refers to the classical activation spectrum of Twomey in the form $N_{CCN}(s) = Cs^k$
  - 'TFH' includes some improvements brought by Feingold and Heymsfield (JAS, 1992) to the original activation spectrum of Twomey.
  - 'CPB' refers to an activation spectrum in the form defined in Cohard et al. (JAS, 1998) with $N_{CCN}(s) = Cs^k F(\mu, \frac{k}{2}, \frac{k}{2}+1; -\beta s^2)$, where F is the hypergeometric function and $[C, k, \mu, \beta]$, four adjustable coefficients,
- HINI_CCN: If HPARAM_CCN=='CPB' then the initial CCN characteristics are given in the 'CCN' or 'AER' format. In the 'CCN' case, the parameters XCHEN, XKHEN, XMUHEN and XBETAHEN must be given while it is the case for XCONC_CCN, XR_MEAN_CCN, XLOGSIG_CCN, XFSOLUB_CCN and XACTEMP_CCN if the 'AER' option is chosen.
  - 'CCN' The aerosols are directly characterized by their activation spectrum $N_{CCN}(s)$ in the form Cs^k or $Cs^k F(\mu, \frac{k}{2}, \frac{k}{2}+1; -\beta s^2)$.
  - 'AER' The aerosols are particles which are characterized by a lognormal distribution law in the form: $N/\sqrt{2\pi}\ln(\sigma) exp\Big(-\ln(r/\overline{r})^2/2\ln(\sigma)^2\Big)$, with distribution parameters ($\overline{r}$ is the geometric mean radius, σ the geometric standard deviation and N the total particle number), by their solubility (ϵ_m) and by their activation temperature (T) as described by Cohard et al. (JGR, 2000).
- HTYPE_CCN: Aerosol type ('M' or 'C') if HPARAM_CCN=='CPB' and HINI_CCN=='AER' is chosen.
  - 'M': NaCl composition (large size maritime aerosols)
  - 'C': $(NH_4)_2SO_4$ composition (small size continental aerosols)
- XCHEN: Parameter C in the generic activation spectrum $N_{CCN}(s)$
- XKHEN: Parameter k in the generic activation spectrum $N_{CCN}(s)$
- XMUHEN: Parameter μ in the hypergeometric function of the CPB form of the activation spectrum $N_{CCN}(s)$

- XBETAHEN: Parameter β in the hypergeometric function of the CPB form of the activation spectrum $N_{CCN}(s)$
- XCONC_CCN: aerosol number concentration (N)
- XR_MEAN_CCN: geometric mean radius of the aerosol distribution ($\overline{r}$)
- XLOGSIG_CCN: natural logarithm of the geometric standard deviation of the aerosol distribution ($\ln(\sigma)$)
- XFSOLUB_CCN: Mean solubility of the aerosols (ϵ_m)
- XACTEMP_CCN: Mean air temperature at which activation will occur.
- XALPHAC: First dispersion parameter (α_c) of the γ-distribution law of the cloud droplets ($\gamma_c(D) = \frac{\alpha_c}{\Gamma(\nu_c)} \lambda_c^{\alpha_c \nu_c} D^{\alpha_c \nu_c - 1} exp(-(\lambda_c D)^{\alpha_c})$)
- XNUC: Second dispersion parameter (ν_c) of the γ-distribution law of the cloud droplets
- XALPHAR: First dispersion parameter (α_r) of the γ-distribution law of the rain drops ($\gamma_r(D) = \frac{\alpha_r}{\Gamma(\nu_r)} \lambda_r^{\alpha_r \nu_r} D^{\alpha_r \nu_r - 1} exp(-(\lambda_r D)^{\alpha_r})$)
- XNUR: Second dispersion parameter (ν_r) of the γ-distribution law of the rain drops
- LRAIN: Enables the rain formation (by cloud droplet autoconversion) when it is TRUE
- LSEDC: Cloud droplets are allowed to sediment when it is TRUE
- LACTIT: Activation by radiative cooling is taken into account when it is TRUE

4.2.9 Namelist NAM_PARAM_ICE (option for the mixed phase cloud parameterization ICE3 and ICE4)

Fortran name	Fortran type	default value
LWARM	logical	.TRUE.
CPRISTINE_ICE	4 characters	'PLAT'
LSEDIC	boolean	FALSE

It contains the options for the mixed phase cloud parameterizations used by the model. They are included in the declarative module MODD_PARAM_ICE

- LWARM : When .TRUE. activates the formation of rain by the warm microphysical processes
- CPRISTINE_ICE : Pristine ice crystal type
  - 'PLAT' : plates

  - 'COLU' : columns
  - 'BURO' : bullet rosettes

- LSEDIC : Cloud droplets are allowed to sediment when it is TRUE

4.2.10 Namelist NAM_TURB_CLOUD (mixing length for clouds)

Fortran name	Fortran type	default value
NMODEL_CLOUD	integer	999
CTURBLEN_CLOUD	4 characters	'DELT'
XCOEF_AMPL_SAT	real	5.
XCEI_MIN	real	0.001E-6
XCEI_MAX	real	0.01E-6

- NMODEL_CLOUD : model number where the modification of the mixing length in the clouds is computed,
- CTURBLEN_CLOUD : type of turbulent mixing length in the clouds ('BL89','DELT','DEAR': see CTURBLEN for meaning),
- XCOEF_AMPL_SAT : saturation of the amplification coefficient,
- XCEI_MIN : minimum threshold for the instability index (in kg/kg/m/s, beginning of the amplication),
- XCEI_MAX : maximum threshold for the instability index (in kg/kg/m/s, beginning of the saturation of the amplification).

Diagnostics quantities are written on every synchronuous files (mixing length in clear sky, mixing length modified, amplification coefficient,...) if LTURB_DIAG=.TRUE. in NAM_TURBn.

4.2.11 Namelist NAM_ADVn (scalar advection schemes of model n)

Fortran name	Fortran type	default value
CUVW_ADV_SCHEME	6 characters	'CEN4TH'
CMET_ADV_SCHEME	6 characters	'FCT2ND'
CSV_ADV_SCHEME	6 characters	'FCT2ND'
NLITER	integer	2

It contains the different advection schemes for dynamic variables (u,v and w), scalar meteorological variables (temperature, water substances, TKE) and tracers used by the model n. They are included in the declarative module MODD_ADVn

- CUVW_ADV_SCHEME: Advection scheme used for horizontal and vertical velocities: The following options are possible :

  - 'CEN2ND' 2nd order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'CEN4TH' 4th order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.

- CMET_ADV_SCHEME: Advection scheme used for the following METeorological variables: temperature, water substances and TKE. The following options are possible (see the Scientific Documentation for more details):
  - 'CEN2ND' 2nd order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'CEN4TH' 4th order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'FCT2ND' 2nd order advection scheme CENtred on space and time. It is POSITIVE definite.
  - 'MPDATA' 2nd order advection scheme uncentred on space and time. It is POSITIVE definite.
  - 'PPM_00' PPM advection scheme without constraint
  - 'PPM_01' Monotonic version of PPM. It is POSITIVE definite.

- CSV_ADV_SCHEME: Advection scheme used for the tracer variables. The following options are possible (see the Scientific Documentation for more details):
  - 'CEN2ND' 2nd order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'CEN4TH' 4th order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'FCT2ND' 2nd order advection scheme CENtred on space and time. It is POSITIVE definite.
  - 'MPDATA' 2nd order advection scheme uncentred on space and time. It is POSITIVE definite.
  - 'PPM_00' PPM advection scheme without constraint
  - 'PPM_01' Monotonic version of PPM. It is POSITIVE definite.

- NLITER : number of iterations that the MPDATA is applied. NLITER=1 (donor cell or upstream scheme).

4.2.12 Namelist NAM_CONFn (configration of model n)

Fortran name	Fortran type	default value
LUSERV	logical	TRUE
LUSECI	logical	FALSE
NSV_USER	integer	0

It contains the model configuration parameters specific for the model n. They are included in the module MODD_CONFn.

- LUSERV : Switch to use vapor water (prognostic variable r_v)
  - .TRUE. rv is present
  - .FALSE. rv is not allocated
- LUSECI : Switch to use Pristine Ice (diagnostic variable C_i)
  - .TRUE. C_i is present
  - .FALSE. C_i is not allocated
- NSV_USER : Number of user passive scalar variables
  Caution! Scalar variables needed for the 2-moments microphysical schemes, lagrangian trajectory or the chemistry options are treated automatically by the model and should not be counted here.

4.2.13 Namelist NAM_DYNn (parameters for the dynamics of model n)

Fortran name	Fortran type	default value
XTSTEP	real	60.
CPRESOPT	4 characters	'CRESI'
NITR	integer	4
LITRADJ	logical	FALSE
XRELAX	real	1.
LHORELAX_UVWTH	logical	FALSE
LHORELAX_RV	logical	FALSE
LHORELAX_RC	logical	FALSE
LHORELAX_RR	logical	FALSE
LHORELAX_RI	logical	FALSE
LHORELAX_RS	logical	FALSE
LHORELAX_RG	logical	FALSE
LHORELAX_RH	logical	FALSE
LHORELAX_TKE	logical	FALSE
LHORELAX_SV	array logical	FALSE
LHORELAX_SVC2R2	logical	FALSE
LHORELAX_SVC1R3	logical	FALSE
LHORELAX_SVLG	logical	FALSE
LHORELAX_SVCHEM	logical	FALSE
LHORELAX_SVDST	logical	FALSE
LHORELAX_SVAER	logical	FALSE
LVE_RELAX	logical	FALSE
NRIMX	integer	1
NRIMY	integer	1
XRIMKMAX	real	$1/(100 * 60.)$
XT4DIFF	real	1800.

It contains the dynamics parameters specific for the model n. They are included in the module MODD_DYNn.

- XTSTEP : Time step in seconds. If the model is not the DAD model, XTSTEP is not taken into account but NDTRATIO in NAM_NESTING.
- CPRESOPT : Pressure solver option. 3 choices are implemented in MESONH for the moment (see the Scientific documentation for more details) :
  - 'RICHA' Richardson method preconditionned by the flat cartesian operator
  - 'CGRAD' Generalized pre-conditioned gradient for non-symmetric problems with the same preconditioner
  - 'CRESI' Conjugate Residual method

  If the problem to be solved is flat and cartesian, then the resolution becomes exact and no iteration is performed.

- NITR : Number of iterations for the iterative pressure solver. The value of this parameter depends on the maximum slope of the orography present in the model.
- LITRADJ : Logical to adjust the number of iterations for the iterative pressure solver according to the range of the residual divergence.
- XRELAX : Relaxation coefficient in the Richardson method (CPRESOPT = 'RICHA'). This value can be less than 1 only for very steep orography, in general, the optimal value is equal to 1.
- LHORELAX_UVWTH : Switch for the horizontal relaxation applied on the outermost verticals of the model for U,V,W TH variables.
  - .TRUE. The horizontal relaxation is applied
  - .FALSE. The horizontal relaxation is not applied
- LHORELAX_RV, LHORELAX_RC, LHORELAX_RR, LHORELAX_RI,
  LHORELAX_RS, LHORELAX_RG, LHORELAX_RH, LHORELAX_TKE,
  LHORELAX_SV, LHORELAX_SVCHEM, LHORELAX_SVC2R2,
  LHORELAX_SVC1R3, LHOREAX_SVLG, LHORELAX_SVDST,
  LHORELAX_SVAER, LHORELAX_SVELEC : idem for other variables

  It is more safety to set all the LHORELAX_ values than use the default values which can be modified by the desfm file.
- LVE_RELAX : Switch for the vertical relaxation applied on the outermost verticals of the model.
  - .TRUE. The vertical relaxation is applied
  - .FALSE. The vertical relaxation is not applied
- NRIMX : number of points included in the lateral relaxation area in the x direction.
- NRIMY : number of points included in the lateral relaxation area in the y direction.
- XRIMKMAX : maximum value (in s^{-1}) of the relaxation coefficient for the lateral relaxation area. This value is appllied on all the outermost verticals of the domain. **Caution : this value is also used to relaxe the normal wind for open lbc condition. This relaxation exists in the Carpenter equation even if LHO_RELAX_UVWTH=F.**
- XT4DIFF : characteristic time (e-folding time) for the numerical diffusion of fourth order. (in seconds). Associated to LNUMDIFF in NAM_DYN.

4.2.14 Namelist NAM_LBCn (boundary conditions of model n)

Fortran name	Fortran type	default value
CLBCX	array(2 characters)	2*"CYCL"
CLBCY	array(2 characters)	2*"CYCL"
XCPHASE	real	20.

It contains the parameters needed to specify the lateral boundary conditions for the model n. They are included in the declarative module MODD_LBCn

- CLBCX : represent the type of lateral boundary condition at the left and right boundaries along x (CLBCX(1) and CLBCX(2) respectively). The possible values are :
  - 'CYCL' for cyclic boundary conditions (in this case CLBCX(1)=CLBCX(2)='CYCL')
  - 'OPEN' for open boundary condition (Sommerfeld equation for the normal velocity)
  - 'WALL' for wall boundary condition (zero normal velocity)
- CLBCY : array containing 2 elements: they represent the type of lateral boundary condition at the left and right boundaries along y (CLBCY(1) and CLBCY(2) respectively). The possible values are identical to those for CLBCX.
- XCPHASE : imposed phase velocity of the outgoing gravity waves. This phase velocity can be used in the Sommerfeld equation which gives the temporal evolution of the normal velosity at the open lateral boundary.

4.2.15 Namelist NAM_LUNITn (file names)

Fortran name	Fortran type	default value
CINIFILE	28 characters	'INIFILE'
CCPLFILE	array (28 characters)	JPCPLFILEMAX*"NONE"

It contains the names of the different files used for the initialization of the model n. They are included in the declarative module MODD_LUNITn

- CINIFILE : name of the initial FM-file which contains the field values used as initial state in the present MESONH numerical simulation
- CCPLFILE : name of the FM-files which contains the field values used for the coupling of the outermost MESONH model. No more than JPCPLFILEMAX (=24 for the present version) files can be used in a simulation. These CCPLFILE files' name are only meaningful for the outermost model which finds its boundary conditions from a previously executed run of Meso-NH or another model.

  No constaint are imposed to the coupling file names but they must be temporally ordered

If the coupling files are given by

CCPLFILE(1)= 'F_1' $- - - >$ t_1
CCPLFILE(2)= 'F_2' $- - - >$ t_2
CCPLFILE(3)= 'A_2' $- - - >$ t_3
CCPLFILE(4)= 'A_5' $- - - >$ t_4
CCPLFILE(5)= 'NONE' $- - - >$
... .
CCPLFILE(8)= 'NONE' $- - - >$

then, the instants must satisfy :

$$t_{segment} \leq t_1 < t_2 < t_3 < t_4$$

If it is not the case, the program stops. The case, for which the coupling fields are not time dependent does not require any coupling file because these coupling fields are read in the initial MESONH file of model 1 as the Larger scale fields (LSUM, LSVM, LSWM, LSTHM, LSRVM). More details can be found in the scientific documentation of the model.

4.2.16 Namelist NAM_NUDGINGn (nudging of model n)

Fortran name	Fortran type	default value
LNUDGING	logical	.FALSE.
XTNUDGING	real	21600.

It contains the parameters needed for nudging of U,V,W,TH,Rv fields of model n towards Large Scale values. They are included in the declarative module MODD_NUDGINGn

- LNUDGING : switch to activate nudging for model n.
- XTNUDGING : time scale for nudging towards Large Scale values.

4.2.17 Namelist NAM_PARAMn (parameterizations' names of model n)

Fortran name	Fortran type	default value
CTURB	4 characters	'NONE'
CRAD	4 characters	'NONE'
CCLOUD	4 characters	'NONE'
CDCONV	4 characters	'NONE'

It contains the types of the different parameterizations used by the model n. They are included in the declarative module MODD_PARAMn

- CTURB : gives the type of turbulence scheme used to parameterize the transfers from unresolved scales to resolved scales.
  - CTURB = 'NONE' : no turbulence scheme.

  - CTURB = 'TKEL' : turbulence scheme with a one and a half order closure (i.e. prognostic turbulent kinetic energy (TKE) and diagnostic mixing length).

- CRAD : gives the type of radiative transfer scheme used to parameterize the effects of the solar and infrared radiations.

  - CRAD = 'NONE' then the downward surface fluxes are set to zero
  - CRAD = 'TOPA' : the solar flux is equal to the one at TOP of Atmosphere. The infra-red flux is equal to 300 Wm^{-2}.
  - CRAD = 'FIXE' then the daily evolutions of the downward surface fluxes are prescribed. The temporal evolution is done in the routine PHYS_PARAMn by fixing the hourly value of the infrared and solar fluxes and can be modified for personnal application.
  - CRAD = 'ECMW' the ECMWF radiation scheme code is used.

- CCLOUD : gives the microphysical scheme used to parameterize the different water phases' transformations.

  - CCLOUD = 'NONE' no microphysical scheme. You nevertheless may use vapor if desired (LUSERV= TRUE or FALSE)
  - CCLOUD = 'REVE' only the saturation adjustment is performed in order to create cloud water if saturation is achievd. This liquid water is never transformed in rain water.
  - CCLOUD = 'KESS' a warm Kessler microphysical scheme is employed. It allows the transformations between 3 classes of water: vapor, cloud water and rain.
  - CCLOUD = 'C2R2' a 2-moment warm microphysical scheme according to Cohard and Pinty (2000).
  - CCLOUD = 'KHKO' a 2-moment warm microphysical scheme for LES of Stratocumulus according to Khairoudinov and Kogan (2000).
  - CCLOUD = 'ICE3' a mixed microphysical scheme including ice, snow, and graupel (6 classes of hydrometeors).
  - CCLOUD = 'ICE4' corresponding to ICE3 with hail (7 classes of hydrometeors).

- CDCONV : gives the type of deep convection scheme used to parameterize the effects of unresolved convective clouds.

  - CDCONV = 'NONE' : no convection scheme.
  - CDCONV = 'KAFR' : Kain-Fritsch-Bechtold scheme.

4.2.18 Namelist NAM_PARAM_CONVECTn (options for the convective scheme of model n)

Fortran name	Fortran type	default value
XDTCONV	real	MAX(300.0,XTSTEP)
NICE	integer	1
LREFRESH_ALL	logical	TRUE
LCHTRANS	logical	FALSE
LDEEP	logical	TRUE
LSHAL	logical	FALSE
LDOWN	logical	TRUE
LSETTADJ	logical	FALSE
XTADJD	real	3600
XTADJS	real	10800
LDIAGCONV	logical	FALSE
NENSM	integer	0

It contains the options retained for the deep convection scheme, used by the model n. They are included in the declarative module MODD_PARAM_CONVECTn

- XDTCONV : timestep for call of the convection scheme. It is limited to 300s.
- NICE : flag to include ice proceses in convection scheme (1 = yes, 0 = no ice)
- LREFRESH_ALL : flag to refresh convective columns at every call of the convection scheme.
- LCHTRANS: flag to take into account the convective transport for scalar variables (can only be used with the options CDCONV='KAFR').
- LDEEP : switch to use deep convection.
- LSHAL : switch to use shallow convection.
- LDOWN : switch to use downdrafts in deep convection.
- LSETTADJ : switch to allow user defined adjustment time.
- XTADJD : user defined deep convective adjustment time (if LSETTADJ=TRUE).
- XTADJS : user defined shallow convective adjustment time (if LSETTADJ=TRUE).
- LDIAGCONV : switch to store diagnostic variables in module MODD_DEEP_CONVECTIONn: (CAPE, deep and shallow convective cloud top and base levels, up-and downdraft mass fluxes)
- NENSM : number of additional convective ensemble members for deep convection (presently limited to 3)

4.2.19 Namelist NAM_PARAM_RADn (options for the radiations of model n)

Fortran name	Fortran type	default value
XDTRAD	real	XTSTEP
XDTRAD_CLONLY	real	XTSTEP
CLW	4 characters	'RRTM'
CAER	4 characters	'SURF'
CEFRADL	4 characters	'MART'
CEFRADI	4 characters	'LIOU'
COPWLW	4 characters	'SMSH'
COPILW	4 characters	'EBCU'
COPWSW	4 characters	'FOUQ'
COPISW	4 characters	'EBCU'
LCLEAR_SKY	logical	FALSE
NRAD_COLNBR	integer	1000
NRAD_DIAG	integer	0
XFUDG	real	1.

It contains the options retained for the radiations scheme, used by the model n. They are included in the declarative module MODD_PARAM_RADn.

- XDTRAD : Interval of time (in seconds) between two full radiation computations. (the radiative tendency is computed for all the verticals of the simulation domain). This is done in order to save CPU time because the radiation scheme is very expensive but the radiative tendency does not evolve too much, in some cases, during periods greater than the model timestep XTSTEP. In this case, the "radiation timestep" is increased to XDTRAD

- XDTRAD_CLONLY : Interval of time (in seconds) between two radiation computations for the cloudy columns only. This is based on the same principle as the intermittent full radiation call: the cloudy column radiative tendency may, in some cases, evolve faster than the dry ones but still slower than the timestep XTSTEP. In this case, the "cloudy radiation timestep" is increased from XDTRAD to XDTRAD_CLONLY. Of course, when all and part of the radiative tendencies must be refreshed during the same MESONH timestep, only the full radiation call is performed.

- CLW : choice of LW code

  - 'RRTM': RAPID RADIATIVE TRANSFER MODEL
  - 'MORC': MORCRETTE model

- CAER : type of aerosol distribution

  - 'SURF': deduced from cover data

  - 'TEGE': computed from Tegen et al. (1997) mensual climatology (horizontal resolution is 4^o for latitude and 5^o for longitude
  - 'TANR': computed from ECMWF T5 climatology
  - 'NONE': no aerosol

- CEFRADL : liquid effective radius calculation
  - 'MART' : based on Martin et al. (1994, JAS)
  - 'C2R2' : based on the prediction of the number concentrations. Recommended for the 2-moment microphysical schemes.
  - 'PRES' : very old parametrization as f(pressure)
  - 'OCLN' : simple distinction between land (10) and ocean (13)

- CEFRADI : ice water effective radius calculation
  - 'LIOU' : ice particle effective radius =f(T) from Liou and Ou (1994)
  - 'SURI' : ice particle effective radius =f(T,IWC) from Sun and Rikus (1999)
  - 'C3R5' : based on the prediction of the number concentrations. Recommended for the 2-moment microphysical schemes (not yet available for mixed clouds).
  - 'FX40' : fixed 40 micron effective radius

- COPWLW : cloud water LW optical properties
  - 'SMSH': Smith-Shi formulation
  - 'SAVI': Savijarvi formulation (only recommended with 2-moment microphysical schemes with small precipitation)

- COPILW : ice water LW optical properties
  - 'EBCU': Ebert-Curry formulation
  - 'SMSH': Smith-Shi formulation, only with CLW='RRTM'
  - 'FULI': Fu-Liou formulation, only with CLW='MORC'

- COPWSW : cloud water SW optical properties
  - 'FOUQ': Fouquart, 1991 formulation
  - 'SLIN': Slingo, 1989 formulation

- COPISW : ice water SW optical properties

  - 'EBCU': Ebert-Curry formulation
  - 'FULI': Fu-Liou formulation

- LCLEAR_SKY : When this logical switch is set to .TRUE., the radiation computations are made for one mean clear-sky and for the whole cloudy columns. This is still the way to spare some CPU time, by postulating that the clear sky columns do not lead to very different radiative tendencies. This hypothesis is only valid in academical cases.

- NRAD_COLNBR : Maximal number of air columns called in a single call of the radiation subroutine. This is performed in order to save memory, because the radiation subroutine allocate for every column of size NKMAX , NKMAX working arrays . This leads to a quadratic dependency of the memory with the number of vertical levels of the model. A way to limit the necessary memory is to split the number of columns passed to the radiation subroutine in several sets of NRAD_COLNBR column. Finally, all the desired columns (depending on the preceding parameters) will be treated by sequentially calling the radiation subroutine for every set of column.

- NRAD_DIAG : number of diagnostic fields related to the radiative scheme stored on every output synchronous files (same fields as NRAD_3D in DIAG program, p.184).

- XFUDG : subgrid cloud inhomogeneity factor.

The cloud overlap assumption is defined in the routine ini_radconf.f90. The different assumptions are :

- $NOVLP = 5$: Random overlap for Clear Sky fraction and Effective Zenithal Angl. It is the best choice without subgrid condensation.

- $NOVLP = 6$: Maximum Random Overlap for Clear Sky fraction, and Random Overlap for Effective Zenithal Angl (DEFAULT VALUE). This option is well adapted to multi-layer clouds.

- $NOVLP = 7$: Maximum overlap for Clear Sky fraction and Random Overlap for Effective Zenithal Angl. This option is well adapted in the absence of multi-layer clouds.

- $NOVLP = 8$: Maximum Random overlap for Clear Sky fraction and Effective Zenithal Angl. It corresponds to the previous configurations before masdev4_7.

4.2.20 Namelist NAM_TURBn (turbulence parameters for model n)

Fortran name	Fortran type	default value
XIMPL	real	1.
CTURBLEN	4 characters	'BL89'
CTURBDIM	4 characters	'1DIM'
LTURB_FLX	logical	FALSE
LTURB_DIAG	logical	FALSE
LSUBG_COND	logical	FALSE
LSUBG_AUCV	logical	FALSE
LSIGMAS	logical	TRUE
LSIG_CONV	logical	FALSE
LRMC01	logical	FALSE
CTOM	4 characters	'NONE'

It contains the characteristics of the turbulence scheme used by the model n. They are included in the declarative module MODD_TURBn

- XIMPL: degree of implicitness of the vertical part of the turbulence scheme. (XIMPL = 0.5 corresponds to the Cranck-Nicholson scheme for the vertical turbulent diffusion and 0. to a purely explicit scheme)
- CTURBDIM: turbulence dimensionnality.
  - CTURBDIM= '1DIM' Only the vertical turbulent fluxes are taken into account. This has to be done for relatively large horizontal meshes.
  - CTURBDIM= '3DIM' All the turbulent fluxes are computed, this is necessary for small horizontal meshes (meso-γ scales or LES)
- CTURBLEN: type of turbulent mixing length.
  - CTURBLEN='DELT' If CTURBDIM='3DIM', the cubic root of the grid volum is used in 3D simulations and the squared root of the volum in 2D simulations. If CTURBB='1DIM', we take Δz in simulation of any dimensionality. This length is always limited by $\kappa * z$ near the ground.
  - CTURBLEN='BL89' The mixing length is computed according to the Bougeault and Lacarrére scheme (refer to the scientific documentation)
  - CTURBLEN='DEAR' the mixing length is given by the mesh size depending on the model dimensionality, this length is limited with the ground distance and also by the Deardorff mixing length pertinent in the stable cases.
- LTURB_FLX: logical switch to compute and store all the turbulent fluxes on every output synchronous files.

- LTURB_DIAG: logical switch to store diagnostic quantities related to the turbulent scheme on every output synchronous files. (mesh length, Prandtl number, Schmidt number, sources of TKE...)
- LSUBG_COND: switch to activate the subgrid condensation scheme (refer to the scientific documentation for more details)
- LSUBG_AUCV: switch to activate the subgrid autoconversion scheme (if LSUBG_COND is set to TRUE and only with the mixed phase for the moment)
- LSIGMAS: Switch for using Sigma_s from turbulence scheme instead parameterized values in ice subgrid condensation scheme
- LSIG_CONV: Switch for computing Sigma_s due to convection in ice subgrid condensation scheme
- LRMC01: Switch for computing separate mixing and dissipative length in the SBL according to Redelsperger, Mahe and Carlotti 2001
- CTOM: Consideration of Third Order Moments.
  - CTOM='NONE': No Third Order moments
  - CTOM='TM06': Parameterization of Third Order moments of heat fluxes for dry CBL according to Tomas and Masson (2006).

4.2.21 SURFACE SCHEMES: namelists of the externalized surface

The further definition of the surface parameters are not done by MESONH itself, but by the externalized surface included in it. So you are invited to refer to the **documentation of the surface**.

To summarize, the following namelists must be defined (version 1 of the externalized surface):
physical schemes for sea and for vegetation

- NAM_SEAFLUXn
- NAM_ISBAn

Chemical schemes for each surface

- NAM_CH_CONTROLn

- NAM_CH_SURFn
- NAM_CH_SEAFLUXn
- NAM_CH_WATFLUXn
- NAM_CH_ISBAn
- NAM_CH_TEBn

Online computation of diagnostics

- NAM_DIAG_SURF_ATMn
- NAM_DIAG_SURFn
- NAM_DIAG_ISBAn
- NAM_DIAG_TEBn

4.2.22 CHEMISTRY scheme : Namelist NAM_CH_MNHCn (control of MNHC)

Fortran name	Fortran type	default value
LUSECHEM	logical	FALSE
LCH_INIT_FIELD	logical	FALSE
LCH_SURFACE_FLUX	logical	FALSE
LCH_CONV_SCAV	logical	FALSE
LCH_EXPLICIT_SCAV	logical	FALSE
LCH_CONV_LINOX	logical	FALSE
CCHEM_INPUT_FILE	80 characters	'EXSEG1.nam'
CCH_TDISCRETIZATION	10 characters	'SPLIT'
NCH_SUBSTEPS	integer	1
LCH_TUV_ONLINE	logical	TRUE
CCH_TUV_LOOKUP	80 characters	'PHOTO.TUV39'
CCH_TUV_CLOUDS	4 characters	'NONE'
XCH_TUV_ALBNEW	real	-1.
XCH_TUV_DOBNEW	real	-1.
XCH_TUV_TUPDATE	real	600.
CCH_VEC_METHOD	3 characters	'MAX'
NCH_VEC_LENGTH	integer	1000
XCH_TS1D_TSTEP	real	600.
CCH_TS1D_COMMENT	80 characters	'no comment'
CCH_TS1D_FILENAME	80 characters	'IO1D'

- LUSECHEM: switch to activate chemistry.
- LCH_INIT_FIELD: switch to activate initialization subroutine CH_INIT_FIELD.

- LCH_SURFACE_FLUX: switch to activate chemical surface fluxes (subroutine CH_SURFACE_FLUX.)
- LCH_CONV_SCAV: switch to activate scavenging of chemical species (gazeous or aerosol) and dusts by convective precipitations
- LCH_EXPLICIT_SCAV: not yet implemented
- LCH_CONV_LINOX: switch to activate the production of NOx by LIghtning flashes inside deep convective clouds and its transport (LCHTRANS must be set to TRUE). If
  - LUSECHEM=.F. : a scalar variable named LINOX are written in the LFI file
  - LUSECHEM=.T. : the convective source is added to the NO chemical variable.
- CCHEM_INPUT_FILE: name of the general purpose input file.
- CCH_TDISCRETIZATION: temporal discretization
  - CCH_TDISCRETIZATION='SPLIT': use time-splitting, input fields for solver arescalar variables at t+dt (derived from XRSVS)
  - CCH_TDISCRETIZATION='CENTER': use centered tendencies, input fields for solver are scalar variables at t (XSVT)
  - CCH_TDISCRETIZATION='LAGGED': use lagged tendencies, input fields for solver are scalar variables at t-dt (XSVM)
- NCH_SUBSTEPS: number of steps to be taken by the solver during **two** time steps of MesoNH; the time ste p of the solver is thus equal to 2*XTSTEP/NCH_SUBSTEPS
- LCH_TUV_ONLINE: switch to activate online photolysis rates calculations (only for 1D simulation). If false, photolysis rates are pre_calculated as a function of s olar zenith angle and surface albedo and interpolated on the model grid.
- CCH_TUV_LOOKUP: name of the lookup table file
- CCH_TUV_CLOUDS: method for calculating the impact of clouds on UV radiations (only for 3-D versi on)
  - CCH_TUV_CLOUDS='NONE' : No cloud correction on UV radiations
  - CCH_TUV_CLOUDS='CHAN' : Cloud correction on UV radiations following Chang et al., [1987]
- XCH_TUV_ALBNEW: surface albedo for photolysis rates calculations (only for 1-D version. For 3-D version, albedos are prescribed as a function of the surface characteristics).

- XCH_TUV_DOBNEW: scaling factor for ozone column dobson
- XCH_TUV_TUPDATE: update frequency to refresh photolysis rates.
- CCH_VEC_METHOD: type of vectorization mask
  - 'MAX' take NCH_VEC_LENGTH points
  - 'TOT' take all grid points
  - 'HOR' take horizontal layers
  - 'VER' take vertical columns
- NCH_VEC_LENGTH: number of points for 'MAX' option
- XCH_TS1D_TSTEP: time between two call to write_ts1d
- CCH_TS1D_COMMENT: comment for write_ts1d
- CCH_TS1D_FILENAME: filename for write_ts1d files

4.2.23 Namelist NAM_CH_ORILAM

This namelist is to activate ORILAM chemical aerosols (lognormal distribution for Aitken and accumulation mode). This parameterization include coagulation (intra and inter modal), nucleation, sedimentation, condensation/adsorption of gas phase. This parameterization need to be run together with gas chemical phase (namelist NAM_CH_MNHCn). For correct representation it is recommended to have in the chemical scheme severals compounds as HNO3 (nitric acid), H2SO4 (or SULF; sulphates), NH3 (ammonium) and CO (carbon monoxyde).

Fortran name	Fortran type	default value
LORILAM	logical	FALSE
LVARSIGI	logical	FALSE
LVARSIGJ	logical	FALSE
LSEDIMAERO	logical	FALSE
LTHERMIJ	logical	FALSE
XINIRADIUSI	real	0.01
XINIRADIUSJ	real	0.5
CRGUNIT	character (len=4)	'MASS'
XINISIGI	real	1.60
XINISIGJ	real	1.60
XN0IMIN	real	10.
XN0JMIN	real	1.
XCOEFRADIMAX	real	10.
XCOEFRADJMAX	real	10.
XCOEFRADIMIN	real	.1
XCOEFRADJMIN	real	.1
CMINERAL	character (len=5)	'NONE'
CORGANIC	character (len=5)	'NONE'
CNUCLEATION	character (len=80)	'NONE'

- LORILAM: switch to activate chemical aerosol (possible if LUSECHEM = .TRUE.).
- LVARSIGI: switch to activate variable standard deviation for mode I (Aitken).
- LVARSIGJ: switch to activate variable standard deviation for mode J (accumulation).
- LSEDIMAERO: switch to activate aerosol sedimentation.
- LTHERMIJ: switch to compute gas/particles thermodynamic balance with separate modes (if TRUE) or for all aerosol mass (if FALSE).
- XINIRADIUSI: to switch on the initialization of mean radius mode I (Aitken mode) of the distribution (in micrometers).
- XINIRADIUSJ: to switch on the initialization of mean radius mode J (accumulation mode) of the distribution (in micrometers).
- CRGUNIT: type of mean radius given in namelist. Default is for a mass spectral distribution; XINIRADIUSI and XINIRADIUSJ has been converted into a mean radius in number.
  IF CRGUNIT =/ 'MASS' then the mean radius need to be given for a number spectral distribution (no conversion).
- XINISIGI: value of standard deviation for mode I (Aitken mode).

- XINISIGJ: value of standard deviation for mode J (accumulation mode).

- XCOEFRADIMAX: factor to compute maximum value of mean radius mode I (Aitken mode). $R_i^{max} = XCOEFRADIMAX.XINIRADIUSI$

- XCOEFRADJMAX: factor to compute maximum value of mean radius mode J (accumulation mode). $R_j^{max} = XCOEFRADJMAX.XINIRADIUSJ$

- XCOEFRADIMIN: factor to compute minimum value of mean radius mode I (Aitken mode). $R_i^{max} = XCOEFRADIMIN.XINIRADIUSI$

- XCOEFRADJMIN: factor to compute minimum value of mean radius mode J (accumulation mode). $R_j^{max} = XCOEFRADJMIN.XINIRADIUSJ$

- CMINERAL: type of parameterization for mineral gas/particle balance. Possible values are:
  CMINERAL = 'ARES' : ARES parameterization (non vectorized)
  CMINERAL = 'NARES': neuronal network of ARES (vectorized)
  CMINERAL = 'ISPIA': ISORROPIA parameterization (non vectorized)
  CMINERAL = 'TABUL': tabulation of ISORROPIA (vectorized)
  CMINERAL = 'EQSAM': EQSAM parameterization (vectorized)

- CORGANIC: type of parameterization for organic gas/particle balance. To activate organic parameterization it is necessary to use a chemical scheme capable to form secondary organic aerosol (i.e. RELACS2 or CACM). Possible values are:
  CORGANIC = 'PUN' : PUN parameterization
  CORGANIC = 'MPMPO': MPMPO (non vectorized)

- CNUCLEATION: type of parameterization for nucleation (formation of new particle from sulphates). Possible values are:
  CNUCLEATION = 'KULMALA' : KULMALA parameterization
  CNUCLEATION = 'KERMINEN': KERMINEN parameterization

- Convective scavenging is activated with LCH_CONV_SCAV in NAM_CH_MNHCn.

4.2.24 Namelist NAM_CH_SOLVERn (control stiff solvers for modeln)

Fortran name	Fortran type	default value
CSOLVER	32 characters	'SIS'
NSSA	integer	0
NSSAINDEX	1000 integers	1000*0
XRTOL	real	0.001
XATOL	real	0.1
NRELAB	integer	2
NPED	integer	1
NMAXORD	integer	5
LPETZLD	logical	TRUE
CMETHOD	1 character	N
CNORM	1 character	A
NTRACE	integer	0
XALPHA	real	0.5
XSLOW	real	100.0
XFAST	real	0.1
NQSSAITER	integer	1
XDTMIN	real	0.1
XDTMAX	real	600.
XDTFIRST	real	10.

4.2.25 Simulation of the fly of balloons or aircraft in the model fields.

In order to compare the model outputs to airborne observations and measurements, it can be interesting to simulate the movement of a balloon or an aircraft during the model run (in any one model for gridnesting runs). A balloon is launch at a given location, and either for a particular density (iso-density balloon), a particular volume (constant volume balloon) or ascent speed (radio-sounding). For iso-density balloons, initial altitude or pressure is asked. A balloon is advected by the wind of the model. It can crash. For an aircraft, the flight legs must be given by the user (location and duration).

All the prognostic fields (zonal and meridien wind (from U and V components), vertical velocity, potential temperature, pression, mixing ratios, tke, radiative surface temperature) are recorded on the trajectory of the balloon or the aircraft, as well as the trajectory itself (position in X, Y and Z directions and orography). All records are in the diachronic file (.000). Up to 9 balloons and aircraft are possible.

The specification of the characteristics of flights are not given in a namelist, but directly in Fortran routines:

- *ini_balloon.f90* for balloons,
- *ini_aircraft.f90* for aircraft.

4.2.26 Profilers and stations

In order to compare the model outputs to observations and measurements, it can be interesting to store the simulated data for a given profiler or station. A profiler is located at a given location with its latitude and longitude, whereas a station is located with its latitude, longitude and altitude. Pronostic fields are recorded at a time frequency prescribed: zonal and meridien wind (from U and V components), vertical velocity, potential temperature, pression, mixing ratios, tke, radiative surface temperature. If surface diagnostics are asked (see surface namelists), surface variables (10m-wind, 2m-temperature and humidity, surface fluxes) are also stored. All records are in the diachronic file (.000).

The specification of the characteristics of profilers and stations are not given in a namelist, but directly in Fortran routines:

- *ini_profilern.f90* for profilers,
- *ini_stationn.f90* for stations.

4.2.27 Namelist NAM_SERIES (temporal series in diagnostic file)

Fortran name	Fortran type	default value
LSERIES	boolean	FALSE
LMASKLANDSEA	boolean	FALSE
LWMINMAX	boolean	FALSE

- LSERIES : switch to write temporal series in the diachronic file (.000) of each model: evolution of horizontaly and verticaly averaged fields (t), evolution of horizontaly averaged vertical profils (z,t), evolution of y-horizontaly averaged fields at one level or verticaly averaged between 2 levels (x,t).
- LMASKLANDSEA : switch to separate sea and land points in temporal series (t) and (z,t),
- LWMINMAX : switch to compute minimum and maximum of vertical velocity W in temporal serie (t).

See also the namelist NAM_SERIESn.

Some examples of temporal series are available which treat pronostic fields averaged or not vertically. If one wants other fields (for example diagnostic fields such as relative humidity), follows extrema, the following Fortran routines must be modified:

- *ini_series.f90* for initialization of size and name of diachronic records,

- *seriesn.f90* to store and eventually vertically average values during the run,

- *write_seriesn.f90* to horizontally average and write series in diachronic file.

4.2.28 Namelist NAM_SERIESn (temporal series in diagnoctic file of model n)

Fortran name	Fortran type	default value
NIBOXL	integer	2
NIBOXH	integer	3
NJBOXL	integer	2
NJBOXH	integer	3
NKCLS	integer	2
NKCLA	integer	2
NKLOW	integer	2
NKMID	integer	2
NKUP	integer	2
NBJSLICE	integer	1
NJSLICEL	array (20 integer)	20*2
NJSLICEH	array (20 integer)	20*3
NFREQSERIES	integer	XSEGLEN/(100*60)

- NIBOXL, NIBOXH, NJBOXL, NJBOXH: lower and upper indexes along x and y axes of the horizontal box used to average the series (t) and (z,t).

- NKCLS, NKCLA: K level respectively in the CLS and CLA ((x,t) series of U, Rv, Rr at KCLS and W at KCLA are stored).

- NKLOW, NKUP: two K levels ((x,t) series of mean W between KLOW and KUP and mean Rc between the ground and KUP are stored).

- NKMID: a K level ((x,t) serie of Rv at KMID is stored).

- NBJSLICE: number of y-slices for (x,t) serie.

- NJSLICEL, NJSLICEH: lower and higher index along y axe of the y-slices.

- NFREQSERIES : Time frequency of diagnostic writing.

4.2.29 Namelist NAM_BUDGET (budget box description)

Fortran name	Fortran type	default value
CBUTYPE	4 characters	'NONE'
NBUMOD	integer	1
XBULEN	real	43200.
NBUKL	integer	1
NBUKH	integer	0
LBU_KCP	logical	TRUE
XBUWRI	real	43200.
NBUIL	integer	1
NBUIH	integer	0
NBUJL	integer	1
NBUJH	integer	0
LBU_ICP	logical	TRUE
LBU_JCP	logical	TRUE
NBU_MASK	integer	1

It contains the description of the box in which the budget are performed. This box is always built with a subset of points of the simulation box.

- CBUTYPE: type of box used to compute the budget:
  - 'CART' a cartesian box defined by the lowest and highest values of the indices in the 3 directions in the MESONH grid, defined in the following.
  - 'MASK' several areas, described by horizontal masks, are selected according to criteria evaluated at each model timestep. The budget computations are realized at the selected verticals for each criteria. The criteria are to be defined in the routine *set_mask.f90*
- NBUMOD: number of the model in which the budget are performed. Only one model must be selected even if the grid-nesting is active.
- NBUMASK: Number of masks used to select the budgets' areas, in the case CBUTYPE= 'MASK'.
- XBULEN: Duration in seconds, on which the different source terms of all the budget are temporally averaged (the minimum value is 2∗ XTSTEP).
- XBUWRI: Duration in seconds, between successive writings in the diachronic file of the budget storage arrays (CBUTYPE='MASK').
- NBUKL: value of the model level K for the bottom of the budget box, in the case of a cartesian box (CBUTYPE='CART').

- NBUKH: value of the model level K for the top of the budget box, in the case of a cartesian box (CBUTYPE='CART'). Inside the budget box:

$$NBUKL \leq K \leq NBUKH$$

- NBUJL: value of the model level J for the left side of the budget box, in the case of a cartesian box (CBUTYPE='CART').

- NBUJH: value of the model level J for the right side of the budget box, in the case of a cartesian box (CBUTYPE='CART'). Inside the budget box:

$$NBUJL \leq J \leq NBUJH$$

- NBUIL: value of the model level I for the left side of the budget box, in the case of a cartesian box (CBUTYPE='CART').

- NBUIH: value of the model level I for the right side of the budget box, in the case of a cartesian box (CBUTYPE='CART'). Inside the budget box:

$$NBUIL \leq J \leq NBUIH$$

- LBU_KCP: Logical switch to average or not in the K direction all the budget terms, for any CBUTYPE value.

- LBU_JCP: Logical switch to average or not in the J direction all the budget terms, for CBUTYPE='CART'.

- LBU_ICP: Logical switch to average or not in the I direction all the budget terms, for CBUTYPE='CART'.

The description of the budgets for every prognostic variable is given below. Because all the budgets are performed in the same way, we give here some details on the way to select or cumulate the different source terms.

Firstly, there is a switch to activate or not the budget of a given prognostic variable. It should be noted that the budget terms for the variable Ψ have the dimension of

$$\frac{\partial \left[\tilde{\rho}\Psi\right]}{\partial t}$$

Then, all the source terms computed in the model for this prognostic variable can be selected according to the following rules:

- NSOURCE_TERM= 0 if you do not want to take into account this source term, it happens when you are only interested in some terms of a budget and not to the whole budget and you discard the others.

- NSOURCE_TERM= 1 if you select this source term and it is the first element of a set of source terms, which are cumulated.

- NSOURCE_TERM= 2 if you select this source term but it is not the first element of a set of source terms, which are cumulated. Note that the source terms which come just after the last of this set of source term, must be discarded (NSOURCE_TERM= 0) or be the first element of a new cumul (NSOURCE_TERM= 1).

We now give every namelist associated to a given prognostic variable.

4.2.30 Namelist NAM_BU_RU (budget for U)

Fortran name	Meaning	Fortran type	default value
LBU_RU	budget switch	logical	FALSE
NASSEU	time filter (Asselin)	integer	0
NNESTU	nesting	integer	0
NADVXU	advection along x	integer	0
NADVYU	advection along y	integer	0
NADVZU	advection along z	integer	0
NFRCU	forcing	integer	0
NNUDU	nudging	integer	0
NCURVU	curvature terms	integer	0
NCORU	Coriolis term	integer	0
NDIFU	numerical diffusion	integer	0
NRELU	relaxation	integer	0
NVTURBU	vert. turb. diffusion	integer	0
NHTURBU	hori. turb. diffusion	integer	0
NPRESU	pressure term	integer	0

4.2.31 Namelist NAM_BU_RV (budget for V)

Fortran name	Meaning	Fortran type	default value
LBU_RV	budget switch	logical	FALSE
NASSEV	time filter (Asselin)	integer	0
NNESTV	nesting	integer	0
NADVXV	advection along x	integer	0
NADVYV	advection along y	integer	0
NADVZV	advection along z	integer	0
NFRCV	forcing	integer	0
NNUDV	nudging	integer	0
NCURVV	curvature terms	integer	0
NCORV	Coriolis term	integer	0
NDIFV	numerical diffusion	integer	0
NRELV	relaxation	integer	0
NHTURBV	hori. turb. diffusion	integer	0
NVTURBV	vert. turb. diffusion	integer	0
NPRESV	pressure term	integer	0

4.2.32 Namelist NAM_BU_RW (budget for W)

Fortran name	Meaning	Fortran type	default value
LBU_RW	budget switch	logical	FALSE
NASSEW	time filter (Asselin)	integer	0
NNESTW	nesting	integer	0
NADVXW	advection along x	integer	0
NADVYW	advection along y	integer	0
NADVZW	advection along z	integer	0
NFRCW	forcing	integer	0
NNUDW	nudging	integer	0
NCURVW	curvature terms	integer	0
NCORW	Coriolis term	integer	0
NGRAVW	gravity term	integer	0
NDIFW	numerical diffusion	integer	0
NRELW	relaxation	integer	0
NHTURBW	hori. turb. diffusion	integer	0
NVTURBW	vert. turb. diffusion	integer	0
NPRESW	pressure term	integer	0

4.2.33 Namelist NAM_BU_RTH (budget for TH)

Fortran name	Meaning	Fortran type	default value
LBU_RTH	budget switch	logical	FALSE
NASSETH	time filter (Asselin)	integer	0
NNESTTH	nesting	integer	0
NADVTH	total advection	integer	0
NADVXTH	advection along x	integer	0
NADVYTH	advection along y	integer	0
NADVZTH	advection along z	integer	0
NFRCTH	forcing	integer	0
NNUDTH	nudging	integer	0
NPREFTH	ref. pressure term	integer	0
NDIFTH	numerical diffusion	integer	0
NRELTH	relaxation	integer	0
NRADTH	radiation	integer	0
NDCONVTH	convection	integer	0
NHTURBTH	hori. turb. diffusion	integer	0
NVTURBTH	vert. turb. diffusion	integer	0
NDISSHTH	dissipation	integer	0
NNEGATH	negative	integer	0
NREVATH	rain evaporation	integer	0
NCONDTH	vapor condensation or cloud water evaporation	integer	0
NHENUTH	heterog. nucleation	integer	0
NHONTH	homogen. nucleation	integer	0
NSFRTH	spontaneous freezing	integer	0
NDEPSTH	deposition of snow	integer	0
NDEPGTH	deposition of graupel	integer	0
NRIMTH	riming of cloud	integer	0
NACCTH	accretion of rain	integer	0
NCFRZTH	conversion freezing	integer	0
NWETGTH	wet growth of graupel	integer	0
NDRYGTH	dry growth of graupel	integer	0
NGMLTTH	graupel melting	integer	0
NIMLTTH	ice melting	integer	0
NBERFITH	bergeron-findeisen	integer	0
NCDEPITH	cond-deposition ice	integer	0
NWETHTH	wet growth of hail	integer	0
NHMLTTH	melting of hail	integer	0

4.2.34 Namelist NAM_BU_RTKE (budget for TKE)

Fortran name	Meaning	Fortran type	default value
LBU_RTKE	budget switch	logical	FALSE
NASSETKE	time filter (Asselin)	integer	0
NADVTKE	total advection	integer	0
NADVXTKE	advection along x	integer	0
NADVYTKE	advection along y	integer	0
NADVZTKE	advection along z	integer	0
NFRCTKE	forcing	integer	0
NDIFTKE	numerical diffusion	integer	0
NRELTKE	relaxation	integer	0
NDPTKE	dynamic production	integer	0
NTPTKE	thermal production	integer	0
NDISSTKE	dissipation of TKE	integer	0
NTRTKE	turbulent transport	integer	0

4.2.35 Namelist NAM_BU_RRV (budget for vapor)

Fortran name	Meaning	Fortran type	default value
LBU_RRV	budget switch	logical	FALSE
NASSERV	time filter (Asselin)	integer	0
NNESTRV	nesting	integer	0
NADVRV	total advection	integer	0
NADVXRV	advection along x	integer	0
NADVYRV	advection along y	integer	0
NADVZRV	advection along z	integer	0
NFRCRV	forcing	integer	0
NNUDRV	nudging	integer	0
NDIFRV	numerical diffusion	integer	0
NRELRV	relaxation	integer	0
NDCONVRV	deep convection	integer	0
NHTURBRV	hori. turb. diffusion	integer	0
NVTURBRV	vert. turb. diffusion	integer	0
NREVARV	rain evaporation	integer	0
NCONDRV	vapor condensation or cloud water evaporation	integer	0
NHENURV	heterogenous nucleation	integer	0
NDEPSRV	deposition on snow	integer	0
NDEPGRV	deposition on graupel	integer	0
NCDEPIRV	cond./deposition on ice	integer	0

4.2.36 Namelist NAM_BU_RRC (budget for cloud water)

Fortran name	Meaning	Fortran type	default value
LBU_RRC	budget switch	logical	FALSE
NASSERC	time filter (Asselin)	integer	0
NNESTRC	nesting	integer	0
NADVRC	total advection	integer	0
NADVXRC	advection along x	integer	0
NADVYRC	advection along y	integer	0
NADVZRC	advection along z	integer	0
NFRCRC	forcing	integer	0
NDIFRC	numerical diffusion	integer	0
NRELRC	relaxation	integer	0
NDCONVRC	deep convection	integer	0
NHTURBRC	hori. turb. diffusion	integer	0
NVTURBRC	vert. turb. diffusion	integer	0
NACCRRC	accretion	integer	0
NAUTORC	autoconversion into rain	integer	0
NSEDIRC	sedimentation of cloud	integer	0
NCONDRC	vapor condensation or cloud water evaporation	integer	0
NHONRC	homogeneous nucleation (ICE3 or ICE4)	integer	0
NRIMRC	riming of cloud water (ICE3 or ICE4)	integer	0
NWETGRC	wet growth of graupel (ICE3 or ICE4)	integer	0
NDRYGRC	dry growth of graupel (ICE3 or ICE4)	integer	0
NIMLTRC	ice melting	integer	0
NBERFIRC	Bergeron-Findeisen gth.	integer	0
NCDEPIRC	cond./deposition on ice	integer	0
NHENURC	CCN activation (C2R2 or KHKO)	integer	0
NSEDIRC	sedimentation (C2R2 or KHKO)	integer	0
NWETHRC	wet growth of hail	integer	0

4.2.37 Namelist NAM_BU_RRR (budget for rain water)

Fortran name	Meaning	Fortran type	default value
LBU_RRR	budget switch	logical	FALSE
NASSERR	time filter (Asselin)	integer	0
NNESTRR	nesting	integer	0
NADVRR	total advection	integer	0
NADVXRR	advection along x	integer	0
NADVYRR	advection along y	integer	0
NADVZRR	advection along z	integer	0
NFRCRR	forcing	integer	0
NDIFRR	numerical diffusion	integer	0
NRELRR	relaxation	integer	0
NACCRRR	accretion of rain droplets	integer	0
NAUTORR	autoconversion into rain droplets	integer	0
NREVARR	rain evaporation	integer	0
NSEDIRR	sedimentation of rain droplets	integer	0
NSFRRR	spontaneous freezing (ICE3 or ICE4)	integer	0
NACCRR	accretion of rain water (ICE3 or ICE4)	integer	0
NCFRZRR	conversion freezing (ICE3 or ICE4)	integer	0
NWETGRR	wet growth of graupel (ICE3 or ICE4)	integer	0
NDRYGRR	dry growth of graupel (ICE3 or ICE4)	integer	0
NGMLTRR	graupel melting(ICE3 or ICE4)	integer	0
NWETHRR	wet growth of hail (ICE4)	integer	0
NHMLTRR	melting of hail (ICE4)	integer	0

4.2.38 Namelist NAM_BU_RRI (budget for non-precipitating ice)

Fortran name	Meaning	Fortran type	default value
LBU_RRI	budget switch	logical	FALSE
NASSERI	time filter (Asselin)	integer	0
NNESTRI	nesting	integer	0
NADVRI	total advection	integer	0
NADVXRI	advection along x	integer	0
NADVYRI	advection along y	integer	0
NADVZRI	advection along z	integer	0
NFRCRI	forcing	integer	0
NDIFRI	numerical diffusion	integer	0
NRELRI	relaxation	integer	0
NDCONVRI	deep convection	integer	0
NHTURBRI	hori. turb. diffusion	integer	0
NVTURBRI	vert. turb. diffusion	integer	0
NNEGARI	negative correction	integer	0
NSEDIRI	sedimentation (ICE3 or ICE4)	integer	0
NHENURI	heterogenous nucleation (ICE3 or ICE4)	integer	0
NHONRI	homogeneous nucleation (ICE3 or ICE4)	integer	0
NAGGSRI	aggregation of snow (ICE3 or ICE4)	integer	0
NAUTSRI	autoconversion of ice (ICE3 or ICE4)	integer	0
NCFRZRI	conversion freezing (ICE3 or ICE4)	integer	0
NWETGRI	wet growth of graupel (ICE3 or ICE4)	integer	0
NDRYGRI	dry growth of graupel (ICE3 or ICE4)	integer	0
NIMLTRI	ice melting (ICE3 or ICE4)	integer	0
NBERFIRI	Bergeron-Findeisen gth. (ICE3 or ICE4)	integer	0
NCDEPIRI	cond./deposition on ice (ICE3 or ICE4)	integer	0
NWETHRI	wet growth of hail (ICE4)	integer	0

4.2.39 Namelist NAM_BU_RRS (budget for snow)

Fortran name	Meaning	Fortran type	default value
LBU_RRS	budget switch	logical	FALSE
NASSERS	time filter (Asselin)	integer	0
NNESTRS	nesting	integer	0
NADVRS	total advection	integer	0
NADVXRS	advection along x	integer	0
NADVYRS	advection along y	integer	0
NADVZRS	advection along z	integer	0
NFRCRS	forcing	integer	0
NDIFRS	numerical diffusion	integer	0
NRELRS	relaxation	integer	0
NNEGARS	negative correction	integer	0
NSEDIRS	sedimentation (ICE3 or ICE4)	integer	0
NDEPSRS	deposition on snow (ICE3 or ICE4)	integer	0
NAGGSRS	aggregation of snow (ICE3 or ICE4)	integer	0
NAUTSRS	autoconversion of ice (ICE3 or ICE4)	integer	0
NRIMRS	riming of cloudwater (ICE3 or ICE4)	integer	0
NACCRS	accretion of rainwater (ICE3 or ICE4)	integer	0
NCMELRS	conversion MeLTing (ICE3 or ICE4)	integer	0
NWETGRS	wet growth of graupel (ICE3 or ICE4)	integer	0
NDRYGRS	dry growth of graupel (ICE3 or ICE4)	integer	0
NWETHRS	wet growth of hail ICE4	integer	0

4.2.40 Namelist NAM_BU_RRG (budget for graupel)

Fortran name	Meaning	Fortran type	default value
LBU_RRG	budget switch	logical	FALSE
NASSERG	time filter (Asselin)	integer	0
NNESTRG	nesting	integer	0
NADVRG	total advection	integer	0
NADVXRG	advection along x	integer	0
NADVYRG	advection along y	integer	0
NADVZRG	advection along z	integer	0
NFRCRG	forcing	integer	0
NDIFRG	numerical diffusion	integer	0
NRELRG	relaxation	integer	0
NNEGARG	negative correction	integer	0
NSEDIRG	sedimentation	integer	0
NSFRRG	spontaneous freezing	integer	0
NDEPGRG	deposition on snow	integer	0
NRIMRG	riming of cloud water	integer	0
NCMELRG	conversion melting	integer	0
NCFRZRG	conversion freezing	integer	0
NWETGRG	wet growth of graupel	integer	0
NDRYGRG	dry growth of graupel	integer	0
NGMLTRG	graupel melting	integer	0
NWETHRG	wet growth of hail (ICE4)	integer	0

4.2.41 Namelist NAM_BU_RRH (budget for hail)

Fortran name	Meaning	Fortran type	default value
LBU_RRH	budget switch	logical	FALSE
NASSERH	time filter (Asselin)	integer	0
NNESTRH	nesting	integer	0
NADVRH	total advection	integer	0
NADVXRH	advection along x	integer	0
NADVYRH	advection along y	integer	0
NADVZRH	advection along z	integer	0
NFRCRH	forcing	integer	0
NDIFRH	numerical diffusion	integer	0
NRELRH	relaxation	integer	0
NNEGARH	negative correction	integer	0
NSEDIRH	sedimentation	integer	0
NWETGRH	wet growth of graupel	integer	0
NWETHRH	wet growth of hail	integer	0
NHMLTRH	hail melting	integer	0

4.2.42 Namelist NAM_BU_RSV (budget for a Scalar Variable)

Fortran name	Meaning	Fortran type	default value
LBU_RSV	budget switch	logical	FALSE
NASSESV	time filter (Asselin)	integer	0
NNESTSV	nesting	integer	0
NADVSV	total advection	integer	0
NADVXSV	advection along x	integer	0
NADVYSV	advection along y	integer	0
NADVZSV	advection along z	integer	0
NFRCSV	forcing	integer	0
NDIFSV	numerical diffusion	integer	0
NRELSV	relaxation	integer	0
NDCONVSV	deep convection	integer	0
NHTURBSV	hori. turb. diffusion	integer	0
NVTURBSV	vert. turb. diffusion	integer	0
NCHEMSV	chemistry activity	integer	0

4.2.43 LES budgets : Namelist NAM_LES

This namelist controls the diagnostics of turbulence, especially for Large Eddy Simulations. The diagnostics are registered in the diachronic file (.000). The list of the diagnostics is given in the following section.

Fortran name	Fortran type	default value
LLES_MEAN	logical	.FALSE.
LLES_RESOLVED	logical	.FALSE.
LLES_SUBGRID	logical	.FALSE.
LLES_UPDRAFT	logical	.FALSE.
LLES_DOWNDRAFT	logical	.FALSE.
LLES_SPECTRA	logical	.FALSE.
NLES_LEVELS	integer (:)	all levels
XLES_HEIGHTS	real (:)	none
NSPECTRA_LEVELS	integer (:)	none
XSPECTRA_HEIGHTS	real (:)	none
NLES_TEMP_SERIE_I	integer(:)	none
NLES_TEMP_SERIE_J	integer(:)	none
XLES_TEMP_SERIE_Z	real(:)	none
CLES_NORM_TYPE	character (len=4)	'NONE'
CBL_HEIGHT_DEF	character (len=3)	'KE '
XLES_TEMP_SAMPLING	real	60 s if CTURB='3DIM' 300 s if CTURB='1DIM'
XLES_TEMP_MEAN_START	real	none
XLES_TEMP_MEAN_END	real	none
XLES_TEMP_MEAN_STEP	real	3600 s
LLES_CART_MASK	logical	.FALSE.
NLES_IINF	integer	physical domain boundary (JPHEXT+1)
NLES_ISUP	integer	physical domain boundary (NIMAX+JPHEXT)
NLES_JINF	integer	physical domain boundary (JPHEXT+1)
NLES_JSUP	integer	physical domain boundary (NJMAX+JPHEXT)
LLES_NEB_MASK	logical	.FALSE.
LLES_CORE_MASK	logical	.FALSE.
LLES_MY_MASK	logical	.FALSE.

- LLES_MEAN: logical switch for computation of the mean vertical profiles of the model variables

- LLES_RESOLVED: logical switch for computation of the mean vertical profiles of the resolved fluxes, variances and covariances

- LLES_SUBGRID: logical switch for computation of the mean vertical profiles of the subgrid fluxes, variances and covariances

- LLES_UPDRAFT: logical switch for computation of the updraft vertical profiles of some resolved and subgrid fluxes, variances and covariances

- LLES_DOWNDRAFT: logical switch for computation of the downdraft vertical profiles of some resolved and subgrid fluxes, variances and covariances

- LLES_SPECTRA: logical switch for computation of the non-local diagnostics (2 points correlations and spectra)
- NLES_LEVELS: list of model levels where the local quantities are computed. Default is: all model levels (per default, the vertical profiles are computed on the MESO-NH grid).
- XLES_HEIGHTS: list of constant altitude levels where the local quantities are computed. Not used per default.
- NSPECTRA_LEVELS: list of model levels where the non-local quantities are computed. Any number is allowed, but too many will be costly in CPU time and memory.
- XSPECTRA_HEIGHTS: list of constant altitude levels where the non-local quantities are computed. Any number is allowed, but too many will be costly in CPU time and memory.
- NLES_TEMP_SERIE_I: list of the I coordinates (model index) of the points where temporal series are extracted. Not yet implemented.
- NLES_TEMP_SERIE_J: list of the J coordinates (model index) of the points where temporal series are extracted. Not yet implemented.
- NLES_TEMP_SERIE_Z: list of the altitudes of the points where temporal series are extracted. Not yet implemented.
- CLES_NORM_TYPE: type of normalization of the fluxes and variances:
  - 'NONE': no normalization is computed (however, the quantities necessary to perform these are computed, and stored in the file)
  - 'CONV': convective normalisation, using Q_0, w_*, h, $< \overline{w'r'_v} >_{surf}$.
  - 'EKMA': Ekman normalization, using u_* and L_{Ekman}.
  - 'MOBU': Monin-Obukhov normalization, using L_{MO}, u_*, Q_0, $< \overline{w'r'_v} >_{surf}$.
- CBL_HEIGHT_DEF: definition of the Boundary Layer height h:
  - 'KE ': test on total kinetic energy: $E(h) + e(h) = 0.05\frac{1}{h}\int_0^h (E(z) + e(z))dz$
  - 'WTV': test on $< w'\theta'_v + \overline{w'\theta'_v} >$: height h where this flux is most negative.
  - 'DTH' : test on θ profile.
- XLES_TEMP_SAMPLING: time (seconds) between two samplings of the LES profiles and non-local quantities
- XLES_TEMP_MEAN_START: time (seconds) from the begining of the simulation at which the averaging begins. If not defined, no averaging is performed.

- XLES_TEMP_MEAN_END: time (seconds) from the begining of the simulation at which the averaging ends. If not defined, no averaging is performed.

- XLES_TEMP_MEAN_STEP: time step (seconds) for averaging.

- LLES_CART_MASK: logical switch to compute the LES diagnostics only inside a cartesian subdomain. Both local and non-local quantities can be computed.

- NLES_IINF: lower i index of the cartesian subdomain. Default is physical domain left boundary.

- NLES_ISUP: upper i index of the cartesian subdomain. Default is physical domain right boundary.

- NLES_JINF: lower j index of the cartesian subdomain. Default is physical domain bottom boundary.

- NLES_JSUP: upper j index of the cartesian subdomain. Default is physical domain top boundary.

- LLES_NEB_MASK: logical switch to compute the LES diagnostics separately inside and outside the model columns where clouds are present. Only local quantities can be computed.

- LLES_CORE_MASK: logical switch to compute the LES diagnostics separately inside and outside the model columns where cloud core is present. Only local quantities can be computed.

- LLES_MY_MASK: logical switch to compute the LES diagnostics on a mask defined by the user. The mask is 2D horizontal. It must be coded at the beginning of the LES monitor routine. Only local quantities can be computed on this mask.

4.2.44 LES budgets : Namelist NAM_PDF

Fortran name	Fortran type	default value
LLES_PDF	logical	.FALSE.
NPDF	integer	none
XTH_PDF_MIN	real	none
XTH_PDF_MAX	real	none
XW_PDF_MIN	real	none
XW_PDF_MAX	real	none
XTHV_PDF_MIN	real	none
XTHV_PDF_MAX	real	none
XRV_PDF_MIN	real	none
XRV_PDF_MAX	real	none
XRC_PDF_MIN	real	none
XRC_PDF_MAX	real	none
XRR_PDF_MIN	real	none
XRR_PDF_MAX	real	none
XRI_PDF_MIN	real	none
XRI_PDF_MAX	real	none
XRS_PDF_MIN	real	none
XRS_PDF_MAX	real	none
XRG_PDF_MIN	real	none
XRG_PDF_MAX	real	none
XRT_PDF_MIN	real	none
XRT_PDF_MAX	real	none
XTHL_PDF_MIN	real	none
XTHL_PDF_MAX	real	none

Each PDF includes $NPDF$ intervals number between X_PDF_MIN and X_PDF_MAX.

- LLES_PDF : Logical switch for pdf computation
- NPDF : Number of PDF intervals
- XTH_PDF_MIN : Minimum of potential temperature pdf
- XTH_PDF_MAX : Maximum of potential temperature pdf
- XW_PDF_MIN : Minimum of vertical velocity pdf
- XW_PDF_MAX : Maximum of vertical velocity pdf
- XTHV_PDF_MIN : Minimum of virtual potential temperature pdf
- XTHV_PDF_MAX : Maximum of virtual potential temperature pdf
- XRV_PDF_MIN : Minimum of vapor mixing ratio pdf
- XRV_PDF_MAX : Maximum of vapor mixing ratio pdf

- XRC_PDF_MIN : Minimum of cloud mixing ratio pdf
- XRC_PDF_MAX : Maximum of cloud mixing ratio pdf
- XRR_PDF_MIN : Minimum of rain mixing ratio pdf
- XRR_PDF_MAX : Maximum of rain mixing ratio pdf
- XRI_PDF_MIN : Minimum of ice mixing ratio pdf
- XRI_PDF_MAX : Maximum of ice mixing ratio pdf
- XRS_PDF_MIN : Minimum of snow mixing ratio pdf
- XRS_PDF_MAX : Maximum of snow mixing ratio pdf
- XRG_PDF_MIN : Minimum of graupel mixing ratio pdf
- XRG_PDF_MAX : Maximum of graupel mixing ratio pdf
- XRT_PDF_MIN : Minimum of total mixing ratio pdf
- XRT_PDF_MAX : Maximum of total mixing ratio pdf
- XTHL_PDF_MIN : Minimum of θ_l pdf
- XTHL_PDF_MAX : Maximum of θ_l pdf

4.3 LES diagnostics

4.3.1 notations

α'	subgrid fluctuation of α	
$\overline{\alpha}$	mean value of α in the grid: resolved quantity	3D
$<\alpha>$	horizontal mean value of α	1D
$\tilde{\alpha} = \overline{\alpha} - <\alpha>$	resolved fluctuation to the mean profile	3D
$<\alpha>_{up}$	horizontal mean value of α in updrafts; only point with $\overline{w}$ greater than $<w>$ are considered	1D
$<\alpha>_{down}$	horizontal mean value of α in downdrafts; only point with $\overline{w}$ smaller than $<w>$ are considered	1D

Examples:

$\alpha'\beta'$		subgrid flux or (co)variance $\alpha'\beta'$	3D
$\overline{\alpha}$		mean value of α in each grid mesh	3D
$\overline{\alpha'}$	$=0$	mean value of the turbulent fluctuation in each grid mesh	3D
$\overline{\alpha'\beta'}$		mean value in each grid mesh of subgrid flux or (co)variance	3D
$\tilde{\alpha}\,\tilde{\beta}$		resolved flux or (co)variance in each grid mesh	3D
$<\alpha>$		horizontal mean value of α	1D
$<\alpha'>$	$=0$	horizontal mean value of a subgrid fluctuation	1D
$<\overline{\alpha}>$	$=<\alpha>$	horizontal mean value of a resolved field	1D
$<\tilde{\alpha}>$	$=0$	horizontal mean value of a resolved fluctuation	1D
$<\overline{\alpha'\beta'}>$		horizontal mean value of subgrid flux or (co)variance	1D
$<\tilde{\alpha}\,\tilde{\beta}>$		horizontal mean value of resolved flux or (co)variance	1D

4.3.2 What is available

The computed fields have usually at least two dimensions: z and t, that is they are temporal evolutions of vertical profiles. They are always written in the diachronic file. Each field have its own group name, say 'NAME'.

When time averaging is asked for, the fields are temporally averaged (and so lose their temporal dimension) , and are written under the name 'A_NAME'.

When normalization is asked for, this one is made individually on each vertical profile, for all times. They are written under the name 'E_NAME'.

When both normalization and time averaging are asked for, normalization is made first, and then time averaging. The resulting vertical profiles are written under the name 'H_NAME'.

4.3.3 LES averaged fields (LLES_MEAN=TRUE)

field	notation in the diachronic file	dim.	general conditions	comments
$< u >$	MEAN_U	z,t,p		dimension 'p' is equal to the
$< v >$	MEAN_V	z,t,p		number of masks.
$< w >$	MEAN_W	z,t,p		when this dimension is not present,
$< p >$	MEAN_PRE	z,t,p		the computation is made only
$< \rho >$	MEAN_RHO	z,t,p		on the cartesian mask.
$< \theta >$	MEAN_TH	z,t,p		
$< \theta_l >$	MEAN_THL	z,t,p	r_c	
$< \theta_v >$	MEAN_THV	z,t,p	r_v	
$< r_v >$	MEAN_RV	z,t,p	r_v	
$< r_c >$	MEAN_RC	z,t,p	r_c	
$< r_r >$	MEAN_RR	z,t,p	r_r	
$< r_i >$	MEAN_RI	z,t,p	r_i	
$< r_s >$	MEAN_RS	z,t,p	r_s	
$< r_g >$	MEAN_RG	z,t,p	r_g	
$< r_h >$	MEAN_RH	z,t,p	r_h	
$< s_v >$	MEAN_SV	z,t,p,n	s_v	
$< \sqrt{\overline{u}^2 + \overline{v}^2} >$	MEAN_WIND	z,t,p		different from $\sqrt{< \overline{u} >^2 + < \overline{v} >^2}$!
$< \overline{\rho} \max(\overline{w}, < w >) >$	MEAN_MSFX	z,t,p		mean upward mass flux

4.3.4 LES pdf (LLES_PDF=TRUE)

field	notation in the diachronic file	dim.	general conditions	comments
PDF_θ	PDF_TH	z,t,p,n		dimension 'p' is equal to the
PDF_W	PDF_W	z,t,p,n		number of masks.
PDF_θ_v	PDF_THV	z,t,p,n		dimension 'n' is equal to the
PDF_R_v	PDF_RV	z,t,p,n		number of pdf intervals NPDF
PDF_R_c	PDF_RC	z,t,p,n		
PDF_R_t	PDF_RT	z,t,p,n		
PDF_θ_l	PDF_THL	z,t,p,n		
PDF_R_r	PDF_RR	z,t,p,n		
PDF_R_i	PDF_RI	z,t,p,n		
PDF_R_s	PDF_RS	z,t,p,n		
PDF_R_g	PDF_RG	z,t,p,n		

4.3.5 LES averaged fields (LLES_RESOLVED=TRUE)

field	notation in the diac. file	dim.	general conditions	comments
$<\tilde{u}^2>$	RES_U2	z,t,p		warning, contains both turbulent and gravity wave fields
$<\tilde{v}^2>$	RES_V2	z,t,p		”
$<\tilde{w}^2>$	RES_W2	z,t,p		”
$<\tilde{u}\ \tilde{v}>$	RES_UV	z,t,p		”
$<\tilde{w}\ \tilde{u}>$	RES_WU	z,t,p		”
$<\tilde{w}\ \tilde{v}>$	RES_WV	z,t,p		”
$<\frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)>$	RES_KE	z,t,p		”
$<\tilde{p}^2>$	RES_P2	z,t,p		”
$<\tilde{u}\frac{\partial}{\partial z}\tilde{p}>$	RES_UPZ	z,t,p		”
$<\tilde{v}\frac{\partial}{\partial z}\tilde{p}>$	RES_VPZ	z,t,p		”
$<\tilde{w}\frac{\partial}{\partial z}\tilde{p}>$	RES_WPZ	z,t,p		”
$<\tilde{\theta}\ \theta_v>$	RES_THTV	z,t,p	r_v	”
$<\tilde{\theta_l}\ \tilde{\theta_v}>$	RES_TLTV	z,t,p	r_c	”
$<\tilde{\theta}^2>$	RES_TH2	z,t,p		”
$<\tilde{\theta_l}^2>$	RES_THL2	z,t,p	r_c	”

field	notation in the diac. file	dim.	general conditions	comments
$< \tilde{u}\tilde{\theta} >$	RES_UTH	z,t,p		warning, contains both turbulent and gravity wave fields
$< \tilde{v}\tilde{\theta} >$	RES_VTH	z,t,p		”
$< \tilde{w}\tilde{\theta} >$	RES_WTH	z,t,p		”
$< \tilde{u}\tilde{\theta_l} >$	RES_UTHL	z,t,p	r_c	”
$< \tilde{v}\tilde{\theta_l} >$	RES_VTHL	z,t,p	r_c	”
$< \tilde{w}\tilde{\theta_l} >$	RES_WTHL	z,t,p	r_c	”
$< \tilde{u}\tilde{\theta_v} >$	RES_UTHV	z,t,p	r_v	”
$< \tilde{v}\tilde{\theta_v} >$	RES_VTHV	z,t,p	r_v	”
$< \tilde{w}\tilde{\theta_v} >$	RES_WTHV	z,t,p	r_v	”
$< \tilde{r_v}^2 >$	RES_RV2	z,t,p	r_v	”
$< \tilde{\theta}\tilde{r_v} >$	RES_THRV	z,t,p	r_v	”
$< \tilde{\theta_l}\tilde{r_v} >$	RES_TLRV	z,t,p	r_c	”
$< \tilde{\theta_v}\tilde{r_v} >$	RES_TVRV	z,t,p	r_v	”
$< \tilde{u}\tilde{r_v} >$	RES_URV	z,t,p	r_v	”
$< \tilde{v}\tilde{r_v} >$	RES_VRV	z,t,p	r_v	”
$< \tilde{w}\tilde{r_v} >$	RES_WRV	z,t,p	r_v	”
$< \tilde{r_c}^2 >$	RES_RC2	z,t,p	r_c	”
$< \tilde{\theta}\tilde{r_c} >$	RES_THRC	z,t,p	r_c	”
$< \tilde{\theta_l}\tilde{r_c} >$	RES_TLRC	z,t,p	r_c	”
$< \tilde{\theta_v}\tilde{r_c} >$	RES_TVRC	z,t,p	r_c	”
$< \tilde{u}\tilde{r_c} >$	RES_URC	z,t,p	r_c	”
$< \tilde{v}\tilde{r_c} >$	RES_VRC	z,t,p	r_c	”
$< \tilde{w}\tilde{r_c} >$	RES_WRC	z,t,p	r_c	”
$< \tilde{r_i}^2 >$	RES_RI2	z,t,p	r_i	”
$< \tilde{\theta}\tilde{r_i} >$	RES_THRI	z,t,p	r_i	”
$< \tilde{\theta_l}\tilde{r_i} >$	RES_TLRI	z,t,p	r_i	”
$< \tilde{\theta_v}\tilde{r_i} >$	RES_TVRI	z,t,p	r_i	”
$< \tilde{u}\tilde{r_i} >$	RES_URI	z,t,p	r_i	”
$< \tilde{v}\tilde{r_i} >$	RES_VRI	z,t,p	r_i	”
$< \tilde{w}\tilde{r_i} >$	RES_WRI	z,t,p	r_i	”
$< \tilde{w}\tilde{r_r} >$	RES_WRR	z,t,p	r_r	”
Precipitation flux	INPRR3D	z,t,p	r_r	”
Max Precipitation flux	MAXINPRR3D	z,t,p	r_r	”
Evaporation flux	EVAP3D	z,t,p	r_r	”
$< \tilde{s_v}^2 >$	RES_SV2	z,t,p,n	s_v	”
$< \tilde{\theta}\tilde{s_v} >$	RES_THSV	z,t,p,n	s_v	”
$< \tilde{\theta_l}\tilde{s_v} >$	RES_TLSV	z,t,p,n	s_v	”
$< \tilde{\theta_v}\tilde{s_v} >$	RES_TVSV	z,t,p,n	s_v	”
$< \tilde{u}\tilde{s_v} >$	RES_USV	z,t,p,n	s_v	”
$< \tilde{v}\tilde{s_v} >$	RES_VSV	z,t,p,n	s_v	”
$< \tilde{w}\tilde{s_v} >$	RES_WSV	z,t,p,n	s_v	”
$< \tilde{w}^3 >$	RES_W3	z,t,p	s_v	”
$< \tilde{w}\ \tilde{\theta_l}^2 >$	RES_WTL2	z,t,p	r_c	”
$< \tilde{w}^2\ \tilde{\theta_l} >$	RES_W2TL	z,t,p	r_c	”
$< \tilde{w}\ \tilde{r_v}^2 >$	RES_WRV2	z,t,p	r_v	”
$< \tilde{w}^2\ \tilde{r_v} >$	RES_W2RV	z,t,p	r_v	”

field	notation in diac. file	dim.	general conditions	comments
$< \tilde{w}\ \tilde{\theta_l} \tilde{r_v} >$	RE_WTLRV	z,t,p	r_v	", if r_v and no r_c, replaced by $< \tilde{w}\ \tilde{\theta} \tilde{r_v} >$
$< \tilde{w}\ \tilde{r_c}^2 >$	RES_WRC2	z,t,p	r_c	"
$< \tilde{w}^2\ \tilde{r_c} >$	RES_W2RC	z,t,p	r_c	"
$< \tilde{w}\ \tilde{\theta_l} \tilde{r_c} >$	RE_WTLRV	z,t,p	r_c	"
$< \tilde{w}\ \tilde{r_v} \tilde{r_c} >$	RE_WRVRC	z,t,p	r_c	"
$< \tilde{w}\ \tilde{r_i}^2 >$	RES_WRI2	z,t,p	r_i	"
$< \tilde{w}^2\ \tilde{r_i} >$	RES_W2RI	z,t,p	r_i	"
$< \tilde{w}\ \tilde{\theta_l} \tilde{r_i} >$	RE_WTLRI	z,t,p	r_i	"
$< \tilde{w}\ \tilde{r_v} \tilde{r_i} >$	RE_WRVRI	z,t,p	r_i	"
$< \tilde{w}\ \tilde{s_v}^2 >$	RES_WSV2	z,t,p,n	s_v	"
$< \tilde{w}^2\ \tilde{s_v} >$	RES_W2SV	z,t,p,n	s_v	"
$< \tilde{w}\ \tilde{\theta_l} \tilde{s_v} >$	RE_WTLSV	z,t,p,n	s_v	", if no r_c, replaced by $< \tilde{w}\ \tilde{\theta} \tilde{s_v} >$
$< \tilde{w}\ \tilde{r_v} \tilde{s_v} >$	RE_WRVSV	z,t,p,n	r_v, s_v	"
$< \tilde{\theta_l} \frac{\partial}{\partial z} \tilde{p} >$	RES_TLPZ	z,t,p		", if no r_c, replaced by $< \tilde{\theta} \frac{\partial}{\partial z,t,p} \tilde{p} >$
$< \tilde{r_v} \frac{\partial}{\partial z} \tilde{p} >$	RES_RVPZ	z,t,p	r_v	"
$< \tilde{r_c} \frac{\partial}{\partial z} \tilde{p} >$	RES_RCPZ	z,t,p	r_c	"
$< \tilde{r_i} \frac{\partial}{\partial z} \tilde{p} >$	RES_RIPZ	z,t,p	r_i	"
$< \tilde{s_v} \frac{\partial}{\partial z} \tilde{p} >$	RES_SVPZ	z,t,p,n	s_v	"
$< \tilde{u} \left[\frac{1}{2} (\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) \right] >$	RES_UKE	z,t,p		"
$< \tilde{v} \left[\frac{1}{2} (\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) \right] >$	RES_VKE	z,t,p		"
$< \tilde{w} \left[\frac{1}{2} (\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) \right] >$	RES_WKE	z,t,p		"

4.3.6 LES averaged fields (LLES_SUBGRID=TRUE)

field	notation in diac. file	dim.	general conditions	comments
$<\frac{1}{2}(\overline{u'^2}+\overline{v'^2}+\overline{w'^2})>$	SBG_TKE	z,t,p		
$<\overline{u'^2}>$	SBG_U2	z,t,p		
$<\overline{v'^2}>$	SBG_V2	z,t,p		
$<\overline{w'^2}>$	SBG_W2	z,t,p		
$<\overline{u'v'}>$	SBG_UV	z,t,p		
$<\overline{w'u'}>$	SBG_WU	z,t,p		
$<\overline{w'v'}>$	SBG_WV	z,t,p		
$<\overline{\theta_l'^2}>$	SBG_THL2	z,t,p		if no r_c, replaced by $<\overline{\theta'^2}>$
$<\overline{u'\theta_l'}>$	SBG_UTHL	z,t,p		if no r_c, replaced by $<\overline{u'\theta'}>$
$<\overline{v'\theta_l'}>$	SBG_VTHL	z,t,p		if no r_c, replaced by $<\overline{v'\theta'}>$
$<\overline{w'\theta_l'}>$	SBG_WTHL	z,t,p		if no r_c, replaced by $<\overline{w'\theta'}>$
$<\overline{w'\theta_v'}>$	SBG_WTHV	z,t,p	r_v	
$<\overline{r_t'^2}>$	SBG_RT2	z,t,p	r_v	r_t is for total water
$<\overline{\theta_l'r_t'}>$	SBG_TLRT	z,t,p	r_v	", if no r_c, replaced by $<\overline{\theta'r_v'}>$
$<\overline{u'r_t'}>$	SBG_URT	z,t,p	r_v	r_t is for total water
$<\overline{v'r_t'}>$	SBG_VRT	z,t,p	r_v	r_t is for total water
$<\overline{w'r_t'}>$	SBG_WRT	z,t,p	r_v	r_t is for total water
$<\overline{r_c'^2}>$	SBG_RC2	z,t,p	r_c	
$<\overline{u'r_c'}>$	SBG_URC	z,t,p	r_c	
$<\overline{v'r_c'}>$	SBG_VRC	z,t,p	r_c	
$<\overline{w'r_c'}>$	SBG_WRC	z,t,p	r_c	
$<\overline{u's_v'}>$	SBG_USV	z,t,p,n	s_v	
$<\overline{v's_v'}>$	SBG_VSV	z,t,p,n	s_v	
$<\overline{w's_v'}>$	SBG_WSV	z,t,p,n	s_v	
$<\overline{u'}\left[\frac{1}{2}(\overline{u'^2}+\overline{v'^2}+\overline{w'^2})\right]>$	SBG_UTKE	z,t,p		
$<\overline{v'}\left[\frac{1}{2}(\overline{u'^2}+\overline{v'^2}+\overline{w'^2})\right]>$	SBG_VTKE	z,t,p		
$<\overline{w'}\left[\frac{1}{2}(\overline{u'^2}+\overline{v'^2}+\overline{w'^2})\right]>$	SBG_WTKE	z,t,p		

4.3.7 LES averaged fields (LLES_UPDRAFT=TRUE)

field	notation in the diac. file	dim.	general conditions	comments
$<f_{up}>$	UP_FRAC	z,t		updraft fraction
$<w>_{up}$	UP_W	z,t		from now on, computations are
$<\theta>_{up}$	UP_TH	z,t		made only on the cartesian mask

field	notation in the diac. file	dim.	general conditions	comments
$< \theta_l >_{up}$	UP_THL	z,t	r_c	
$< \theta_v >_{up}$	UP_THV	z,t	r_v	
$< \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) >_{up}$	UP_KE	z,t		
$< \frac{1}{2}(\widetilde{u'^2} + \widetilde{v'^2} + \widetilde{w'^2}) >_{up}$	UP_TKE	z,t		
$< r_v >_{up}$	UP_RV	z,t	r_v	
$< r_c >_{up}$	UP_RC	z,t	r_c	
$< r_r >_{up}$	UP_RR	z,t	r_r	
$< r_i >_{up}$	UP_RI	z,t	r_i	
$< r_s >_{up}$	UP_RS	z,t	r_s	
$< r_g >_{up}$	UP_RG	z,t	r_g	
$< r_h >_{up}$	UP_RH	z,t	r_h	
$< s_v >_{up}$	UP_SV	z,t,n	s_v	
$< \tilde{\theta}^2 >_{up}$	UP_TH2	z,t		
$< \tilde{\theta_l}^2 >_{up}$	UP_THL2	z,t	r_c	
$< \tilde{\theta}\ \tilde{\theta_v} >_{up}$	UP_THTV	z,t	r_v	
$< \tilde{\theta_l}\ \tilde{\theta_v} >_{up}$	UP_TLTV	z,t	r_c	
$< \tilde{w}\ \tilde{\theta} >_{up}$	UP_WTH	z,t		
$< \tilde{w}\ \tilde{\theta_l} >_{up}$	UP_WTHL	z,t	r_c	
$< \tilde{w}\ \tilde{\theta_v} >_{up}$	UP_WTHV	z,t	r_v	
$< \tilde{r_v}^2 >_{up}$	UP_RV2	z,t	r_v	
$< \tilde{\theta}\ \tilde{r_v} >_{up}$	UP_THRV	z,t	r_v	
$< \tilde{\theta_l}\ \tilde{r_v} >_{up}$	UP_TLRV	z,t	r_c	
$< \tilde{\theta_v}\ \tilde{r_v} >_{up}$	UP_TVRV	z,t	r_v	
$< \tilde{w}\ \tilde{r_v} >_{up}$	UP_WRV	z,t	r_v	
$< \tilde{r_c}^2 >_{up}$	UP_RC2	z,t	r_c	
$< \tilde{\theta}\ \tilde{r_c} >_{up}$	UP_THRC	z,t	r_c	
$< \tilde{\theta_l}\ \tilde{r_c} >_{up}$	UP_TLRC	z,t	r_c	
$< \tilde{\theta_v}\ \tilde{r_c} >_{up}$	UP_TVRC	z,t	r_c	
$< \tilde{w}\ \tilde{r_c} >_{up}$	UP_WRC	z,t	r_c	
$< \tilde{r_i}^2 >_{up}$	UP_RI2	z,t	r_i	
$< \tilde{\theta}\ \tilde{r_i} >_{up}$	UP_THRI	z,t	r_i	
$< \tilde{\theta_l}\ \tilde{r_i} >_{up}$	UP_TLRI	z,t	r_i	
$< \tilde{\theta_v}\ \tilde{r_i} >_{up}$	UP_TVRI	z,t	r_i	
$< \tilde{w}\ \tilde{r_i} >_{up}$	UP_WRI	z,t	r_i	
$< \tilde{s_v}^2 >_{up}$	UP_SV2	z,t,n	s_v	
$< \tilde{\theta}\ \tilde{s_v} >_{up}$	UP_THSV	z,t,n		
$< \tilde{\theta_l}\ \tilde{s_v} >_{up}$	UP_TLSV	z,t,n	r_c, s_v	
$< \tilde{\theta_v}\ \tilde{s_v} >_{up}$	UP_TVSV	z,t,n	r_v, s_v	
$< \tilde{w}\ \tilde{s_v} >_{up}$	UP_WSV	z,t,n	s_v	

4.3.8 LES averaged fields (LLES_DOWNDRAFT=TRUE)

field	notation in the diac. file	dim.	general conditions	comments
$< f_{dw} >$	DW_FRAC	z,t		downdraft fraction
$< w >_{dw}$	DW_W	z,t		
$< \theta >_{dw}$	DW_TH	z,t		
$< \theta_l >_{dw}$	DW_THL	z,t	r_c	
$< \theta_v >_{dw}$	DW_THV	z,t	r_v	
$< \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) >_{dw}$	DW_KE	z,t		
$< \frac{1}{2}(\widetilde{u'^2} + \widetilde{v'^2} + \widetilde{w'^2}) >_{dw}$	DW_TKE	z,t		
$< r_v >_{dw}$	DW_RV	z,t	r_v	
$< r_c >_{dw}$	DW_RC	z,t	r_c	
$< r_r >_{dw}$	DW_RR	z,t	r_r	
$< r_i >_{dw}$	DW_RI	z,t	r_i	
$< r_s >_{dw}$	DW_RS	z,t	r_s	
$< r_g >_{dw}$	DW_RG	z,t	r_g	
$< r_h >_{dw}$	DW_RH	z,t	r_h	
$< s_v >_{dw}$	DW_SV	z,t,n	s_v	
$< \tilde{\theta}^2 >_{dw}$	DW_TH2	z,t		
$< \tilde{\theta_l}^2 >_{dw}$	DW_THL2	z,t	r_c	
$< \tilde{\theta}\,\tilde{\theta_v} >_{dw}$	DW_THTV	z,t	r_v	
$< \tilde{\theta_l}\,\tilde{\theta_v} >_{dw}$	DW_TLTV	z,t	r_c	
$< \tilde{w}\,\tilde{\theta} >_{dw}$	DW_WTH	z,t		
$< \tilde{w}\,\tilde{\theta_l} >_{dw}$	DW_WTHL	z,t	r_c	
$< \tilde{w}\,\tilde{\theta_v} >_{dw}$	DW_WTHV	z,t	r_v	
$< \tilde{r_v}^2 >_{dw}$	DW_RV2	z,t	r_v	
$< \tilde{\theta}\,\tilde{r_v} >_{dw}$	DW_THRV	z,t	r_v	
$< \tilde{\theta_l}\,\tilde{r_v} >_{dw}$	DW_TLRV	z,t	r_c	
$< \tilde{\theta_v}\,\tilde{r_v} >_{dw}$	DW_TVRV	z,t	r_v	
$< \tilde{w}\,\tilde{r_v} >_{dw}$	DW_WRV	z,t	r_v	
$< \tilde{r_c}^2 >_{dw}$	DW_RC2	z,t	r_c	
$< \tilde{\theta}\,\tilde{r_c} >_{dw}$	DW_THRC	z,t	r_c	
$< \tilde{\theta_l}\,\tilde{r_c} >_{dw}$	DW_TLRC	z,t	r_c	
$< \tilde{\theta_v}\,\tilde{r_c} >_{dw}$	DW_TVRC	z,t	r_c	
$< \tilde{w}\,\tilde{r_c} >_{dw}$	DW_WRC	z,t	r_c	
$< \tilde{r_i}^2 >_{dw}$	DW_RI2	z,t	r_i	
$< \tilde{\theta}\,\tilde{r_i} >_{dw}$	DW_THRI	z,t	r_i	
$< \tilde{\theta_l}\,\tilde{r_i} >_{dw}$	DW_TLRI	z,t	r_i	
$< \tilde{\theta_v}\,\tilde{r_i} >_{dw}$	DW_TVRI	z,t	r_i	
$< \tilde{w}\,\tilde{r_i} >_{dw}$	DW_WRI	z,t	r_i	
$< \tilde{s_v}^2 >_{dw}$	DW_SV2	z,t,n	s_v	
$< \tilde{\theta}\,\tilde{s_v} >_{dw}$	DW_THSV	z,t,n		
$< \tilde{\theta_l}\,\tilde{s_v} >_{dw}$	DW_TLSV	z,t,n	r_c, s_v	
$< \tilde{\theta_v}\,\tilde{s_v} >_{dw}$	DW_TVSV	z,t,n	r_v, s_v	
$< \tilde{w}\,\tilde{s_v} >_{dw}$	DW_WSV	z,t,n	s_v	

4.3.9 LES averaged surface fields

field	notation in diac. file	dimension	general conditions	comments
$< \overline{w'\theta'}_{surf} >$	Q0	t		surface sensible flux
$< \overline{w'r'}_{vsurf} >$	E0	t	r_v	surface latent flux
$< \overline{w's'}_{vsurf} >$	E0	t,n	s_v	surface scalar flux
$u_* = \left\{ < \overline{u'w'}_{surf} >^2 + < \overline{v'w'}_{surf} >^2 \right\}^{\frac{1}{4}}$	U*	t		friction velocity
$w_* = \left\{ < \frac{g}{\theta} >< \overline{w'\theta'}_{vsurf} >< h > \right\}^{\frac{1}{3}}$	W*	t		convective velocity, if positive surface buoyancy flux
$< h >$	BL_H	t		boundary layer height
$< L_{MO} >$	L_MO	t		Monin-Obukhov length
$\int TKEdz$	INT_TKE	t		vertical integrated TKE
	ZCB	t		cloud base height
CF	ZCFTOT	t	r_c	total cloud cover
$\int \rho(r_c + r_r)$	LWP	t	r_c	Cloud water path
VAR_LWP	LWPVAR	t	r_c	LWP variance
$\int \rho r_r$	RWP	t	r_r	Rain water path
$INPRR$	INST_PREC	t	r_r	Inst. precip. rate
	RAIN_PREC	t	r_r	INPRR over rainy grids
$ACPRR$	ACCU_PREC	t	r_r	Accum. precip. rate
H_CFmax	ZMAXCF	t	r_c	Height of cloud fraction maximun

4.3.10 LES 2 points correlations

field	notation in the diac. file	dim.	general conditions	comments
$< \tilde{u}(x,y)\ \tilde{u}(x+l_x,y) >$	CI_UU	l_x,2,z,t		
$< \tilde{u}(x,y)\ \tilde{u}(x,y+l_y) >$	CJ_UU	l_y,2,z,t		
$< \tilde{v}(x,y)\ \tilde{v}(x+l_x,y) >$	CI_VV	l_x,2,z,t		
$< \tilde{v}(x,y)\ \tilde{v}(x,y+l_y) >$	CJ_VV	l_y,2,z,t		
$< \tilde{w}(x,y)\ \tilde{w}(x+l_x,y) >$	CI_WW	l_x,2,z,t		
$< \tilde{w}(x,y)\ \tilde{w}(x,y+l_y) >$	CJ_WW	l_y,2,z,t		
$< \tilde{u}(x,y)\ \tilde{v}(x+l_x,y) >$	CI_UV	l_x,2,z,t		
$< \tilde{u}(x,y)\ \tilde{v}(x,y+l_y) >$	CJ_UV	l_y,2,z,t		
$< \tilde{w}(x,y)\ \tilde{u}(x+l_x,y) >$	CI_WU	l_x,2,z,t		
$< \tilde{w}(x,y)\ \tilde{u}(x,y+l_y) >$	CJ_WV	l_y,2,z,t		
$< \tilde{\theta}(x,y)\ \tilde{\theta}(x+l_x,y) >$	CI_THTH	l_x,2,z,t		
$< \tilde{\theta}(x,y)\ \tilde{\theta}(x,y+l_y) >$	CJ_THTH	l_y,2,z,t		
$< \tilde{\theta_l}(x,y)\ \tilde{\theta_l}(x+l_x,y) >$	CI_TLTL	l_x,2,z,t	r_c	
$< \tilde{\theta_l}(x,y)\ \tilde{\theta_l}(x,y+l_y) >$	CJ_TLTL	l_y,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{\theta}(x+l_x,y) >$	CI_WTH	l_x,2,z,t		
$< \tilde{w}(x,y)\ \tilde{\theta}(x,y+l_y) >$	CJ_WTH	l_y,2,z,t		
$< \tilde{w}(x,y)\ \tilde{\theta_l}(x+l_x,y) >$	CI_WTHL	l_x,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{\theta_l}(x,y+l_y) >$	CJ_WTHL	l_y,2,z,t	r_c	
$< \tilde{r_v}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_RVRV	l_x,2,z,t	r_v	
$< \tilde{r_v}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_RVRV	l_y,2,z,t	r_v	
$< \tilde{\theta}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_THRV	l_x,2,z,t	r_v	
$< \tilde{\theta}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_THRV	l_y,2,z,t	r_v	
$< \tilde{\theta_l}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_TLRV	l_x,2,z,t	r_c	
$< \tilde{\theta_l}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_TLRV	l_y,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_WRV	l_x,2,z,t	r_v	
$< \tilde{w}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_WRV	l_y,2,z,t	r_v	
$< \tilde{r_c}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_RCRC	l_x,2,z,t	r_c	
$< \tilde{r_c}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_RCRC	l_y,2,z,t	r_c	
$< \tilde{\theta}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_THRC	l_x,2,z,t	r_c	
$< \tilde{\theta}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_THRC	l_y,2,z,t	r_c	
$< \tilde{\theta_l}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_TLRC	l_x,2,z,t	r_c	
$< \tilde{\theta_l}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_TLRC	l_y,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_WRC	l_x,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_WRC	l_y,2,z,t	r_c	
$< \tilde{r_i}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_RIRI	l_x,2,z,t	r_i	
$< \tilde{r_i}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_RIRI	l_y,2,z,t	r_i	
$< \tilde{\theta}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_THRI	l_x,2,z,t	r_i	
$< \tilde{\theta}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_THRI	l_y,2,z,t	r_i	
$< \tilde{\theta_l}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_TLRI	l_x,2,z,t	r_i	
$< \tilde{\theta_l}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_TLRI	l_y,2,z,t	r_i	
$< \tilde{w}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_WRI	l_x,2,z,t	r_i	
$< \tilde{w}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_WRI	l_y,2,z,t	r_i	
$< \tilde{s_v}(x,y)\ \tilde{s_v}(x+l_x,y) >$	CI_SVSV	l_x,2,z,t,n	s_v	
$< \tilde{s_v}(x,y)\ \tilde{s_v}(x,y+l_y) >$	CJ_SVSV	l_y,2,z,t,n	s_v	
$< \tilde{w}(x,y)\ \tilde{s_v}(x+l_x,y) >$	CI_WSV	l_x,2,z,t,n	s_v	
$< \tilde{w}(x,y)\ \tilde{s_v}(x,y+l_y) >$	CJ_WSV	l_y,2,z,t,n	s_v	

4.3.11 LES spectra

field	notation in the diac. file	dim.	general conditions	comments
$\frac{1}{2\pi L}\int_L <\tilde{u}(x,y)\ \tilde{u}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_UU	k_x,2,z,t		dimension 2
$\frac{1}{2\pi L}\int_L <\tilde{u}(x,y)\ \tilde{u}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_UU	k_y,2,z,t		is for real
$\frac{1}{2\pi L}\int_L <\tilde{v}(x,y)\ \tilde{v}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_VV	k_x,2,z,t		and
$\frac{1}{2\pi L}\int_L <\tilde{v}(x,y)\ \tilde{v}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_VV	k_y,2,z,t		imaginary
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{w}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_WW	k_x,2,z,t		parts
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{w}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_WW	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{u}(x,y)\ \tilde{v}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_UV	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{u}(x,y)\ \tilde{v}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_UV	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{u}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_WU	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{u}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_WV	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{\theta}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_THTH	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{\theta}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_THTH	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{\theta_l}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_TLTL	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{\theta_l}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_TLTL	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{\theta}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_WTH	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{\theta}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_WTH	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{\theta_l}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_WTHL	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{\theta_l}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_WTHL	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{r_v}(x,y)\ \tilde{r_v}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_RVRV	k_x,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L <\tilde{r_v}(x,y)\ \tilde{r_v}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_RVRV	k_y,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{r_v}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_THRV	k_x,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{r_v}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_THRV	k_y,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{r_v}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_TLRV	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{r_v}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_TLRV	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{r_v}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_WRV	k_x,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{r_v}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_WRV	k_y,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L <\tilde{r_c}(x,y)\ \tilde{r_c}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_RCRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{r_c}(x,y)\ \tilde{r_c}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_RCRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{r_c}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_THRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{r_c}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_THRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{r_c}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_TLRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{r_c}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_TLRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{r_c}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_WRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{w}(x,y)\ \tilde{r_c}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_WRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L <\tilde{r_i}(x,y)\ \tilde{r_i}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_RIRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L <\tilde{r_i}(x,y)\ \tilde{r_i}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_RIRI	k_y,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{r_i}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_THRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L <\tilde{\theta}(x,y)\ \tilde{r_i}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_THRI	k_y,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{r_i}(x+l_x,y)>\ e^{-ik_x l_x}dl_x$	SI_TLRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L <\tilde{\theta_l}(x,y)\ \tilde{r_i}(x,y+l_y)>\ e^{-ik_y l_y}dl_y$	SJ_TLRI	k_y,2,z,t	r_i	

field	notation in the diac. file	dim.	general conditions	comments
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r_i}(x+l_x,y) > \ e^{-ik_x l_x} dl_x$	SI_WRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r_i}(x,y+l_y) > \ e^{-ik_y l_y} dl_y$	SJ_WRI	k_y,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{s_v}(x,y)\ \tilde{s_v}(x+l_x,y) > \ e^{-ik_x l_x} dl_x$	SI_SVSV	k_x,2,z,t,n	s_v	
$\frac{1}{2\pi L}\int_L < \tilde{s_v}(x,y)\ \tilde{s_v}(x,y+l_y) > \ e^{-ik_y l_y} dl_y$	SJ_SVSV	k_y,2,z,t,n	s_v	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{s_v}(x+l_x,y) > \ e^{-ik_x l_x} dl_x$	SI_WSV	k_x,2,z,t,n	s_v	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{s_v}(x,y+l_y) > \ e^{-ik_y l_y} dl_y$	SJ_WSV	k_y,2,z,t,n	s_v	

4.4 Budget of (resolved + subgrid) turbulent quantities

4.4.1 Budget of total turbulent kinetic energy

All terms of the equation of $\frac{\partial}{\partial t}(E+e)$ are computed and stored in the diachronic group BU_KE. Here, e and E denote the subgrid and resolved Tke respectively:

$$< e >=< \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) > \qquad < E >=< \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) >$$

Here are **main terms** of the equations for the horizontal mean of subgrid Tke and resolved Tke, in the frame of Boussinesq approximation. Note that the computations of the budgets terms are done with the complete equation set and discretization of MESONH. The equations here are simplified only for the sake of easier understanding. Other terms can arise from the parametrizations of MESONH, and will also be taken into account in the budget.

$$\frac{\partial}{\partial t} < e >= \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < e >}^{ADVM} \overbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} e >}^{ADVR} \overbrace{- \frac{1}{< \rho >} < \overline{u'_\alpha \frac{\partial p'}{\partial x_\alpha}} >}^{PRES} \overbrace{- < \overline{u'_\alpha u'_\beta} > \frac{\partial < u_\beta >}{\partial x_\alpha}}^{DPM}$$
$$\underbrace{- < \overline{u'_\alpha u'_\beta} \frac{\partial \tilde{u_\beta}}{\partial x_\alpha} >}_{DPR} \underbrace{+ < \beta \overline{w' \theta'_v} >}_{TP} \underbrace{- < \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha e} >}_{TR} \underbrace{- < \epsilon >}_{DISS}$$

$$\frac{\partial}{\partial t} < E >= \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < E >}^{ADV} \overbrace{- \frac{1}{< \rho >} < \tilde{u_\alpha} \frac{\partial \tilde{p}}{\partial x_\alpha} >}^{PRES} \overbrace{- < \tilde{u_\alpha} \tilde{u_\beta} > \frac{\partial < u_\beta >}{\partial x_\alpha}}^{DP}$$
$$\underbrace{+ < \beta \tilde{w} \tilde{\theta_v} >}_{TP} \underbrace{- \frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha} E >}_{TR} \underbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\beta} \overline{u'_\alpha u'_\beta} >}_{SBGT} + ...$$

The terms of (spectral) transport from resolved to subgrid motions is SBTG in the equation of $< E >$ (sink), and ADVR and DPR in the equation of $< e >$ (sources). One should note that:

$$\text{ADVR} + \text{DPR} = -\text{SGBT}$$

Note in case of grinesting

In case of 2way gridnesting, the subgrid scheme is not alone to influence the resolved motions due to subgrid scale. Part of the job is done by the averaged of the smaller-scale models. The terms of (spectral) transport from resolved to subgrid motions are then both SBTG and NEST in the equation of $< E >$ (sinks). Therefore

$$\mathrm{ADVR} + \hat{\mathrm{DPR}} = -(\mathrm{SGBT} + \mathrm{NEST})$$

Where $\hat{\mathrm{DPR}}$ is the dynamical production that should produce the subgrid-scale model to equilibrate the sink at resolved scale.

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < e >$	BU_KE	SBG TEND	z,t	(opposite of) tendency of $< e >$
$- < \overline{u'w'} > \frac{\partial}{\partial z} < u >$ $- < \overline{v'w'} > \frac{\partial}{\partial z} < v >$	BU_KE	SBG DP M	z,t	dyn. prod. by mean gradients
$< -\overline{u'_\alpha u'_\beta} \frac{\partial}{\partial x_\beta} \tilde{u_\alpha} >$	BU_KE	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$- < w > \frac{\partial}{\partial z} < e >$	BU_KE	SBG ADVM	z,t	advection by mean flow
$- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} e >$	BU_KE	SBG ADVR	z,t	advection by resolved flow
$-W_{forc} \frac{\partial}{\partial z} < e >$	BU_KE	SBG FORC	z,t	advection by large-scale W forcing
$- < \frac{\partial}{\partial z} \overline{w'e} >$	BU_KE	SBG TR	z,t	subgrid turbulent transport
$-\frac{1}{<\rho>} < \overline{u'_\alpha \frac{\partial p'}{\partial x_\alpha}} >$	BU_KE	SBG PRES	z,t	subgrid pressure-correlation term
$< \beta \overline{w'\theta'_v} >$	BU_KE	SBG TP	z,t	thermal production
$- < \epsilon >$	BU_KE	SBG DISS	z,t	dissipation
numerical diffusion of $< e >$	BU_KE	SBG NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< e >$	BU_KE	SBG RELA	z,t	sponge layer relaxation
miscellaneous	BU_KE	SBG MISC	z,t	...
residual of budget of $< e >$	BU_KE	SBG RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < E >$	BU_KE	RES TEND	z,t	(opposite of) tendency of $< E >$
$- < w > \frac{\partial}{\partial z} < E >$	BU_KE	RES ADV	z,t	advection by mean flow
$-W_{forc} \frac{\partial}{\partial z} < E >$	BU_KE	RES FORC	z,t	advection by large-scale W forcing
$- < \tilde{u}\tilde{w} > \frac{\partial}{\partial z} < u >$ $- < \tilde{v}\tilde{w} > \frac{\partial}{\partial z} < v >$	BU_KE	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha} E >$	BU_KE	RES TR	z,t	transport of resolved Tke by itself
$-\frac{1}{<\rho>} < \tilde{u_\alpha} \frac{\partial \tilde{p}}{\partial x_\alpha} >$	BU_KE	RES PRES	z,t	pressure-correlations
$< \beta \tilde{w} \tilde{\theta_v} >$	BU_KE	RES TP	z,t	thermal production
$- < \tilde{u_\alpha} \frac{\partial}{\partial x_\beta} \overline{u'_\alpha u'_\beta} >$	BU_KE	RES SBGT	z,t	sink due to subgrid turbulence
Coriolis terms	BU_KE	RES CORI	z,t	should be zero for $< E >$
numerical diffusion of $< E >$	BU_KE	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< E >$	BU_KE	RES RELA	z,t	sponge layer relaxation
2way nesting of $< E >$	BU_KE	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_KE	RES MISC	z,t	curvature terms, ...
residual of budget of $< E >$	BU_KE	RES RESI	z,t	must be zero

Note that if a term is zero, because the process accounting for it is not activated in the model, the term is not listed in the diachronic file. So, in order to know which terms have been computed and stored, use the command 'print BU_KE proc' in diaprog.

4.4.2 Budget of total (liquid) temperature flux

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{w}\tilde{\theta_l}> + <\overline{w'\theta'_l}>)$ are computed and stored in the diachronic group BU_WTHL. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{w'\theta'_l}> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{w'\theta'_l}>}^{ADVM} \overbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w'\theta'_l}>}^{ADVR} \overbrace{-<\overline{u'_\alpha w'}\frac{\partial<\theta_l>}{\partial x_\alpha}> - <\overline{u'_\alpha\theta'_l}\frac{\partial<w>}{\partial x_\alpha}>}^{DPM}$$
$$\underbrace{-<\overline{u'_\alpha w'}\frac{\partial\tilde{\theta_l}}{\partial x_\alpha}> - <\overline{u'_\alpha\theta'_l}\frac{\partial\tilde{w}}{\partial x_\alpha}>}_{DPR} \underbrace{+<\beta\overline{\theta'_l\theta'_v}>}_{TP} \underbrace{-\frac{1}{<\rho>}<\theta'_l\overline{\frac{\partial p'}{\partial z}}>}_{PRES} \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'\theta'_l}>}_{TR}$$

$$\frac{\partial}{\partial t}<\tilde{w}\tilde{\theta_l}> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{w}\tilde{\theta_l}>}^{ADV} \overbrace{-\frac{1}{<\rho>}<\tilde{\theta_l}\frac{\partial\tilde{p}}{\partial x_\alpha}>}^{PRES} \overbrace{-<\tilde{u_\alpha}\tilde{w}>\frac{\partial<\theta_l>}{\partial x_\alpha} - <\tilde{u_\alpha}\tilde{\theta_l}>\frac{\partial<w>}{\partial x_\alpha}}^{DP}$$
$$\underbrace{+<\beta\tilde{\theta_l}\tilde{\theta_v}>}_{TP} \underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{\theta_l}>}_{TR} \underbrace{-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha\theta'_l}> - <\tilde{\theta_l}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>}_{SBGT} +...$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'^2}>\frac{\partial}{\partial z}<\theta_l>$	BU_WTHL	SBG DP M	z,t	dyn. prod. by mean gradient
$<-\overline{w'^2}\frac{\partial}{\partial z}\tilde{\theta_l}>$	BU_WTHL	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'^2\theta'_l}>$	BU_WTHL	SBG TR	z,t	subgrid turbulent transport
$-\frac{1}{<\rho>}<\overline{\theta'_l\frac{\partial p'}{\partial z}}>$	BU_WTHL	SBG PRES	z,t	subgrid pressure-correlation term
$<\beta\overline{\theta'_l\theta'_v}>$	BU_WTHL	SBG TP	z,t	thermal production
residual of budget of $<\overline{w'\theta'_l}>$	BU_WTHL	SBG RESI	z,t	must be small

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES TEND	z,t	(opposite of) tendency of $< \tilde{w}\tilde{\theta_l} >$
$- < w > \frac{\partial}{\partial z} < \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES FORC	z,t	advection by large-scale W forcing
$- < \tilde{w}^2 > \frac{\partial}{\partial z} < \theta_l >$ $- < \tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < w >$	BU_WTHL	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES TR	z,t	transport of resolved flux by itself
$-\frac{1}{<\rho>} < \tilde{\theta_l}\frac{\partial \tilde{p}}{\partial z} >$	BU_WTHL	RES PRES	z,t	pressure-correlations
$< \beta\tilde{\theta_l}\tilde{\theta_v} >$	BU_WTHL	RES TP	z,t	thermal production
$- < \tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha\theta'_l} >$ $- < \tilde{\theta_l}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'} >$	BU_WTHL	RES SBGT	z,t	sink due to subgrid turbulence
Coriolis terms	BU_WTHL	RES CORI	z,t	
numerical diffusion of $< \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_WTHL	RES MISC	z,t	ref. pressure term, curvature term, microphysics, radiation, ...
residual of budget of $< \tilde{w}\tilde{\theta_l} >$	BU_WTHL	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{w'\theta'_l} >$ (neglected in turb. scheme)	BU_WTHL	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{w'\theta'_l} >$
$- < w > \frac{\partial}{\partial z} < \overline{w'\theta'_l} >$	BU_WTHL	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w'\theta'_l} >$	BU_WTHL	NSG ADVR	z,t	(neglected) advection by resolved flow
terms due to $\overline{w}$ gradients	BU_WTHL	NSG DPGW	z,t	(neglected) dyn. prod. terms
terms due to hor. $\overline{\theta_l}$ gradients	BU_WTHL	NSG DPGT	z,t	other (neglected) dyn. prod. terms

4.4.3 Budget of total (liquid) temperature variance

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{\theta_l}^2> + <\overline{\theta_l'^2}>)$ are computed and stored in the diachronic group BU_THL2. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{\theta_l'^2}> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{\theta_l'^2}>}^{ADVM} \underbrace{-2<\overline{u_\alpha'\theta_l'}\frac{\partial\tilde{\theta_l}}{\partial x_\alpha}>}_{DPR} \overbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{\theta_l'^2}>}^{ADVR} \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u_\alpha'\theta_l'^2}>}_{TR} \overbrace{-2<\overline{u_\alpha'\theta_l'}>\frac{\partial<\theta_l>}{\partial x_\alpha}}^{DPM} \underbrace{-\epsilon_\theta}_{DISS}$$

$$\frac{\partial}{\partial t}<\tilde{\theta_l}^2> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{\theta_l}^2>}^{ADV} \underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{\theta_l}^2>}_{TR} \overbrace{-2<\tilde{u_\alpha}\tilde{\theta_l}>\frac{\partial<\theta_l>}{\partial x_\alpha}}^{DP} \underbrace{-2<\tilde{\theta_l}\frac{\partial}{\partial x_\alpha}\overline{u_\alpha'\theta_l'}>}_{SBGT} +...$$

field	notation in diac. file	processus name	dim.	comments
$-2<\overline{w'\theta_l'}>\frac{\partial}{\partial z}<\theta_l>$	BU_WTHL	SBG DP M	z,t	dyn. prod. by mean gradient
$<-2\overline{w'\theta_l'}\frac{\partial}{\partial z}\tilde{\theta_l}>$	BU_WTHL	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'\theta_l'^2}>$	BU_WTHL	SBG TR	z,t	subgrid turbulent transport
$-<\epsilon_\theta>$	BU_WTHL	SBG DISS	z,t	dissipation
residual of budget of $<\overline{w'\theta_l'}>$	BU_WTHL	SBG RESI	z,t	must be small
$-\frac{\partial}{\partial t}<\tilde{\theta_l}^2>$	BU_WTHL	RES TEND	z,t	(opposite of) tendency of $<\tilde{\theta_l}^2>$
$-<w>\frac{\partial}{\partial z}<\tilde{\theta_l}^2>$	BU_WTHL	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z}<\tilde{\theta_l}^2>$	BU_WTHL	RES FORC	z,t	advection by large-scale W forcing
$-<2\tilde{w}\tilde{\theta_l}>\frac{\partial}{\partial z}<\theta_l>$	BU_WTHL	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{\theta_l}^2>$	BU_WTHL	RES TR	z,t	resolved transport of resolved variance

field	notation in diac. file	processus name	dim.	comments
$- < 2\tilde{\theta}_l \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha \theta'_l} >$	BU_WTHL	RES SBGT	z,t	sink due to subgrid turbulence
numerical diffusion of $< \tilde{\theta}_l^{\,2} >$	BU_WTHL	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{\theta}_l^{\,2} >$	BU_WTHL	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{\theta}_l^{\,2} >$	BU_WTHL	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_WTHL	RES MISC	z,t	ref. pressure term, radiation, microphysics, ...
residual of budget of $< \tilde{\theta}_l^{\,2} >$	BU_WTHL	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{\theta'^2_l} >$ (neglected in turb. scheme)	BU_WTHL	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{\theta'^2_l} >$
$- < w > \frac{\partial}{\partial z} < \overline{\theta'^2_l} >$	BU_WTHL	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{\theta'^2_l} >$	BU_WTHL	NSG ADVR	z,t	(neglected) advection by resolved flow

4.4.4 Budget of total water flux

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{w}\tilde{r_t}> + <\overline{w'r'_t}>)$ are computed and stored in the diachronic group BU_WRT. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{w'r'_t}> = \overbrace{\underbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{w'r'_t}>}_{ADVM}\ \underbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w'r'_t}>}_{ADVR}}^{DPR}\ \underbrace{+<\beta\overline{r'_t\theta'_v}>}_{TP}\ \overbrace{\underbrace{-<\overline{u'_\alpha w'}\frac{\partial\tilde{r_t}}{\partial x_\alpha}>}_{DPM}\ \underbrace{-<\overline{u'_\alpha r'_t}\frac{\partial\tilde{w}}{\partial x_\alpha}>}_{DPM}}^{DPR}\ \underbrace{-\frac{1}{<\rho>}<r'_t\overline{\frac{\partial p'}{\partial z}}>}_{PRES}\ \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'r'_t}>}_{TR}$$

$$\frac{\partial}{\partial t}<\tilde{w}\tilde{r_t}> = \overbrace{\underbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{w}\tilde{r_t}>}_{ADV}\ \underbrace{-\frac{1}{<\rho>}<\tilde{r_t}\frac{\partial\tilde{p}}{\partial x_\alpha}>}_{PRES}}^{DP}\ \underbrace{+<\beta\tilde{r_t}\tilde{\theta_v}>}_{TP}\ \underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{r_t}>}_{TR}\ \underbrace{-<\tilde{u_\alpha}\tilde{w}>\frac{\partial<r_t>}{\partial x_\alpha}-<\tilde{u_\alpha}\tilde{r_t}>\frac{\partial<w>}{\partial x_\alpha}\ -<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t}>-<\tilde{r_t}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>}_{SBGT}\ +...$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'^2}>\frac{\partial}{\partial z}<r_t>$	BU_WRT	SBG DP M	z,t	dyn. prod. by mean gradient
$<-\overline{w'^2}\frac{\partial}{\partial z}\tilde{r_t}>$	BU_WRT	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'^2 r'_t}>$	BU_WRT	SBG TR	z,t	subgrid turbulent transport
$-\frac{1}{<\rho>}<r'_t\frac{\partial p'}{\partial z}>$	BU_WRT	SBG PRES	z,t	subgrid pressure-correlation term
$<\beta\overline{r'_t\theta'_v}>$	BU_WRT	SBG TP	z,t	thermal production
residual of budget of $<\overline{w'r'_t}>$	BU_WRT	SBG RESI	z,t	must be small
$-\frac{\partial}{\partial t}<\tilde{w}\tilde{r_t}>$	BU_WRT	RES TEND	z,t	(opposite of) tendency of $<\tilde{w}\tilde{r_t}>$
$-<w>\frac{\partial}{\partial z}<\tilde{w}\tilde{r_t}>$	BU_WRT	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z}<\tilde{w}\tilde{r_t}>$	BU_WRT	RES FORC	z,t	advection by large-scale W forcing
$-<\tilde{w}^2>\frac{\partial}{\partial z}<r_t>$ $-<\tilde{w}\tilde{r_t}>\frac{\partial}{\partial z}<w>$	BU_WRT	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{r_t}>$	BU_WRT	RES TR	z,t	transport of resolved flux by itself
$-\frac{1}{<\rho>}<\tilde{r_t}\frac{\partial\tilde{p}}{\partial z}>$	BU_WRT	RES PRES	z,t	pressure-correlations
$<\beta\tilde{r_t}\tilde{\theta_v}>$	BU_WRT	RES TP	z,t	thermal production
$-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t}>$ $-<\tilde{r_t}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>$	BU_WRT	RES SBGT	z,t	sink due to subgrid turbulence
Coriolis terms	BU_WRT	RES CORI	z,t	
numerical diffusion of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES RELA	z,t	sponge layer relaxation
2way nesting of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_WRT	RES MISC	z,t	ref. pressure term, curvature term, radiation, microphysics, ...
residual of budget of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t}<\overline{w'r'_t}>$ (neglected in turb. scheme)	BU_WRT	NSG TEND	z,t	(neglected) opposite of tendency of $<\overline{w'r'_t}>$
$-<w>\frac{\partial}{\partial z}<\overline{w'r'_t}>$	BU_WRT	NSG ADVM	z,t	(neglected) advection by mean flow
$-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w'r'_t}>$	BU_WRT	NSG ADVR	z,t	(neglected) advection by resolved flow
terms due to $\overline{w}$ gradients	BU_WRT	NSG DPGW	z,t	(neglected) dyn. prod. terms
terms due to hor. $\overline{r_t}$ gradients	BU_WRT	NSG DPGT	z,t	other (neglected) dyn. prod. terms

4.4.5 Budget of liquid temperature - total water covariance

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{\theta}_l\tilde{r}_t> + <\overline{\theta'_l r'_t}>)$ are computed and stored in the diachronic group BU_THLR. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{\theta'_l r'_t}> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{\theta'_l r'_t}>}^{ADVM} \overbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{\theta'_l r'_t}>}^{ADVR} \overbrace{-<\overline{u'_\alpha\theta'_l}>\frac{\partial<r_t>}{\partial x_\alpha} - <\overline{u'_\alpha r'_t}>\frac{\partial<\theta_l>}{\partial x_\alpha}}^{DPM}$$
$$\underbrace{-<\overline{u'_\alpha\theta'_l}\frac{\partial\tilde{r}_t}{\partial x_\alpha}> - <\overline{u'_\alpha r'_t}\frac{\partial\tilde{\theta}_l}{\partial x_\alpha}>}_{DPR} \quad \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha\theta'_l r'_t}>}_{TR} \quad \underbrace{-\epsilon_{\theta r}}_{DISS}$$

$$\frac{\partial}{\partial t}<\tilde{\theta}_l\tilde{r}_t> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{\theta}_l\tilde{r}_t>}^{ADV} \overbrace{-<\tilde{u_\alpha}\tilde{\theta}_l>\frac{\partial<r_t>}{\partial x_\alpha} - <\tilde{u_\alpha}\tilde{r}_t>\frac{\partial<\theta_l>}{\partial x_\alpha}}^{DP}$$
$$\underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{\theta}_l\tilde{r}_t>}_{TR} \quad \underbrace{-<\tilde{\theta}_l\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t}> - <\tilde{r}_t\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha\theta'_l}>}_{SBGT} \quad +...$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'\theta'_l}>\frac{\partial}{\partial z}<r_t>$ $-<\overline{w'r'_t}>\frac{\partial}{\partial z}<\theta_l>$	BU_THLR	SBG DP M	z,t	dyn. prod. by mean gradient
$<-\overline{u'_\alpha\theta'_l}\frac{\partial}{\partial x_\alpha}\tilde{r}_t>$ $<-\overline{u'_\alpha r'_t}\frac{\partial}{\partial x_\alpha}\tilde{\theta}_l>$	BU_THLR	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'\theta'_l r'_t}>$	BU_THLR	SBG TR	z,t	subgrid turbulent transport
$-<\epsilon_{\theta r}>$	BU_THLR	SBG DISS	z,t	dissipation
residual of budget of $<\overline{w'\theta'_l r'_t}>$	BU_THLR	SBG RESI	z,t	must be small

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES TEND	z,t	(opposite of) tendency of $< \tilde{\theta_l}\tilde{r_t} >$
$- < w > \frac{\partial}{\partial z} < \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES FORC	z,t	advection by large-scale W forcing
$- < \tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < r_t >$ $- < \tilde{w}\tilde{r_t} > \frac{\partial}{\partial z} < \theta_l >$	BU_THLR	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES TR	z,t	resolved transport of resolved flux
$- < \tilde{\theta_l}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t} >$ $- < \tilde{r_t}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha \theta'_l} >$	BU_THLR	RES SBGT	z,t	sink due to subgrid turbulence
numerical diffusion of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_THLR	RES MISC	z,t	ref. pressure term, radiation, microphysics, ...
residual of budget of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{\theta'_l r'_t} >$ (neglected in turb. scheme)	BU_THLR	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{w' r'_t} >$
$- < w > \frac{\partial}{\partial z} < \overline{\theta'_l r'_t} >$	BU_THLR	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{\theta'_l r'_t} >$	BU_THLR	NSG ADVR	z,t	(neglected) advection by resolved flow

4.4.6 Budget of total water variance

All terms of the equation of $\frac{\partial}{\partial t}(< \tilde{r_t}^2 > + < \overline{r_t'^2} >)$ are computed and stored in the diachronic group BU_RT2. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t} < \overline{r_t'^2} > = \overbrace{\underbrace{-< u_\alpha > \frac{\partial}{\partial x_\alpha} < \overline{r_t'^2} > -2 < \overline{u_\alpha' r_t'} \frac{\partial \tilde{r_t}}{\partial x_\alpha} >}_{DPR}}^{ADVM} \overbrace{\underbrace{-< \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{r_t'^2} > - < \frac{\partial}{\partial x_\alpha} \overline{u_\alpha' r_t'^2} >}_{TR}}^{ADVR} \overbrace{\underbrace{-2 < \overline{u_\alpha' r_t'} > \frac{\partial < r_t >}{\partial x_\alpha} - \epsilon_r}_{DISS}}^{DPM}$$

$$\frac{\partial}{\partial t} < \tilde{r_t}^2 > = \overbrace{\underbrace{-< u_\alpha > \frac{\partial}{\partial x_\alpha} < \tilde{r_t}^2 > - \frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha} \tilde{r_t}^2 >}_{TR}}^{ADV} \overbrace{\underbrace{-2 < \tilde{u_\alpha} \tilde{r_t} > \frac{\partial < r_t >}{\partial x_\alpha} -2 < \tilde{r_t} \frac{\partial}{\partial x_\alpha} \overline{u_\alpha' r_t'} >}_{SBGT}}^{DP} + ...$$

field	notation in diac. file	processus name	dim.	comments
$-2 < \overline{w' r_t'} > \frac{\partial}{\partial z} < r_t >$	BU_RT2	SBG DP M	z,t	dyn. prod. by mean gradient
$< -2 \overline{w' r_t'} \frac{\partial}{\partial z} \tilde{r_t} >$	BU_RT2	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$- < \frac{\partial}{\partial z} \overline{w' r_t'^2} >$	BU_RT2	SBG TR	z,t	subgrid turbulent transport
$- < \epsilon_r >$	BU_RT2	SBG DISS	z,t	dissipation
residual of budget of $< \overline{w' r_t'} >$	BU_RT2	SBG RESI	z,t	must be small

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \tilde{r_t}^2 >$	BU_RT2	RES TEND	z,t	(opposite of) tendency of $< \tilde{r_t}^2 >$
$- < w > \frac{\partial}{\partial z} < \tilde{r_t}^2 >$	BU_RT2	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{r_t}^2 >$	BU_RT2	RES FORC	z,t	advection by large-scale W forcing
$- < 2\tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < r_t >$	BU_RT2	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{r_t}^2 >$	BU_RT2	RES TR	z,t	resolved transport of resolved variance
$- < 2\tilde{r_t}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t} >$	BU_RT2	RES SBGT	z,t	sink due to subgrid turbulence
numerical diffusion of $< \tilde{r_t}^2 >$	BU_RT2	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{r_t}^2 >$	BU_RT2	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{r_t}^2 >$	BU_RT2	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_RT2	RES MISC	z,t	ref. pressure term, radiation, microphysics, ...
residual of budget of $< \tilde{r_t}^2 >$	BU_RT2	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{r_t'^2} >$ (neglected in turb. scheme)	BU_RT2	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{r_t'^2} >$
$- < w > \frac{\partial}{\partial z} < \overline{r_t'^2} >$	BU_RT2	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{r_t'^2} >$	BU_RT2	NSG ADVR	z,t	(neglected) advection by resolved flow

4.4.7 Budget of total scalar flux

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{w}\tilde{s_v}> + <\overline{w's'_v}>)$ are computed and stored in the diachronic group BU_WSV. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{w's'_v}> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{w's'_v}>}^{ADVM} \overbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w's'_v}>}^{ADVR} \overbrace{-<\overline{u'_\alpha w'}\frac{\partial<s_v>}{\partial x_\alpha}} - \overbrace{<\overline{u'_\alpha s'_v}\frac{\partial<w>}{\partial x_\alpha}}^{DPM}$$
$$\underbrace{-<\overline{u'_\alpha w'}\frac{\partial\tilde{s_v}}{\partial x_\alpha}> - <\overline{u'_\alpha s'_v}\frac{\partial\tilde{w}}{\partial x_\alpha}>}_{DPR} + \underbrace{<\beta\overline{s'_v\theta'_v}>}_{TP} \underbrace{-\frac{1}{<\rho>}<s'_v\overline{\frac{\partial p'}{\partial z}}>}_{PRES} \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w's'_v}>}_{TR}$$

$$\frac{\partial}{\partial t}<\tilde{w}\tilde{s_v}> = \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{w}\tilde{s_v}>}^{ADV} \overbrace{-\frac{1}{<\rho>}<\tilde{s_v}\frac{\partial\tilde{p}}{\partial x_\alpha}>}^{PRES} \overbrace{-<\tilde{u_\alpha}\tilde{w}>\frac{\partial<s_v>}{\partial x_\alpha} - <\tilde{u_\alpha}\tilde{s_v}>\frac{\partial<w>}{\partial x_\alpha}}^{DP}$$
$$\underbrace{+<\beta\tilde{s_v}\tilde{\theta_v}>}_{TP} \underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{s_v}>}_{TR} \underbrace{-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha s'_v}> - <\tilde{s_v}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>}_{SBGT} +...$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'^2}>\frac{\partial}{\partial z}<s_v>$	BU_WSV	SBG DP M	z,t,n	dyn. prod. by mean gradient
$<-\overline{w'^2}\frac{\partial}{\partial z}\tilde{s_v}>$	BU_WSV	SBG DP R	z,t,n	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'^2 s'_v}>$	BU_WSV	SBG TR	z,t,n	subgrid turbulent transport
$-\frac{1}{<\rho>}<s'_v\frac{\partial p'}{\partial z}>$	BU_WSV	SBG PRES	z,t,n	subgrid pressure-correlation term
$<\beta\overline{s'_v\theta'_v}>$	BU_WSV	SBG TP	z,t,n	thermal production
residual of budget of $<\overline{w's'_v}>$	BU_WSV	SBG RESI	z,t,n	must be small

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \tilde{w}\tilde{s_v} >$	BU_WSV	RES TEND	z,t,n	(opposite of) tendency of $< \tilde{w}\tilde{s_v} >$
$- < w > \frac{\partial}{\partial z} < \tilde{w}\tilde{s_v} >$	BU_WSV	RES ADV	z,t,n	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{w}\tilde{s_v} >$	BU_WSV	RES FORC	z,t,n	advection by large-scale W forcing
$- < \tilde{w}^2 > \frac{\partial}{\partial z} < s_v >$ $- < \tilde{w}\tilde{s_v} > \frac{\partial}{\partial z} < w >$	BU_WSV	RES DP	z,t,n	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{w}\tilde{s_v} >$	BU_WSV	RES TR	z,t,n	transport of resolved flux by itself
$-\frac{1}{<\rho>} < \tilde{s_v}\frac{\partial \tilde{p}}{\partial z} >$	BU_WSV	RES PRES	z,t,n	pressure-correlations
$< \beta\tilde{s_v}\tilde{\theta_v} >$	BU_WSV	RES TP	z,t,n	thermal production
$- < \tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha s'_v} >$ $- < \tilde{s_v}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'} >$	BU_WSV	RES SBGT	z,t,n	sink due to subgrid turbulence
Coriolis terms	BU_WSV	RES CORI	z,t,n	
numerical diffusion of $< \tilde{w}\tilde{s_v} >$	BU_WSV	RES NUMD	z,t,n	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{w}\tilde{s_v} >$	BU_WSV	RES RELA	z,t,n	sponge layer relaxation
2way nesting of $< \tilde{w}\tilde{s_v} >$	BU_WSV	RES NEST	z,t,n	average from smaller nested models
miscellaneous	BU_WSV	RES MISC	z,t,n	curvature term, chemistry, ...
residual of budget of $< \tilde{w}\tilde{s_v} >$	BU_WSV	RES RESI	z,t,n	must be zero
$-\frac{\partial}{\partial t} < \overline{w's'_v} >$ (neglected in turb. scheme)	BU_WSV	NSG TEND	z,t,n	(neglected) opposite of tendency of $< \overline{w's'_v} >$
$- < w > \frac{\partial}{\partial z} < \overline{w's'_v} >$	BU_WSV	NSG ADVM	z,t,n	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w's'_v} >$	BU_WSV	NSG ADVR	z,t,n	(neglected) advection by resolved flow
terms due to $\overline{w}$ gradients	BU_WSV	NSG DPGW	z,t,n	(neglected) dyn. prod. terms
terms due to hor. $\overline{s_v}$ gradients	BU_WSV	NSG DPGT	z,t,n	other (neglected) dyn. prod. terms

4.4.8 Budget of total scalar variance

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{s_v}^2> + <\overline{s_v'^2}>)$ are computed and stored in the diachronic group BU_SV2. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{s_v'^2}> = \overbrace{\underbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{s_v'^2}>}_{}\ \underbrace{-2<\overline{u_\alpha' s_v'}\frac{\partial \tilde{s_v}}{\partial x_\alpha}>}_{DPR}}^{ADVM} \overbrace{\underbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{s_v'^2}>}_{}\ \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u_\alpha' s_v'^2}>}_{TR}}^{ADVR} \overbrace{\underbrace{-2<\overline{u_\alpha' s_v'}>\frac{\partial <s_v>}{\partial x_\alpha}}_{}\ \underbrace{-\epsilon_{s_v}}_{DISS}}^{DPM}$$

$$\frac{\partial}{\partial t}<\tilde{s_v}^2> = \overbrace{\underbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{s_v}^2>}_{}\ \underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{s_v}^2>}_{TR}}^{ADV} \overbrace{\underbrace{-2<\tilde{u_\alpha}\tilde{s_v}>\frac{\partial <s_v>}{\partial x_\alpha}}_{}\ \underbrace{-2<\tilde{s_v}\frac{\partial}{\partial x_\alpha}\overline{u_\alpha' s_v'}>}_{SBGT}}^{DP} +...$$

field	notation in diac. file	processus name	dim.	comments
$-2<\overline{w's_v'}>\frac{\partial}{\partial z}<s_v>$	BU_SV2	SBG DP M	z,t,n	dyn. prod. by mean gradient
$<-2\overline{w's_v'}\frac{\partial}{\partial z}\tilde{s_v}>$	BU_SV2	SBG DP R	z,t,n	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w's_v'^2}>$	BU_SV2	SBG TR	z,t,n	subgrid turbulent transport
$-<\epsilon_{s_v}>$	BU_SV2	SBG DISS	z,t,n	dissipation
residual of budget of $<\overline{w's_v'}>$	BU_SV2	SBG RESI	z,t,n	must be small

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \tilde{s_v}^2 >$	BU_SV2	RES TEND	z,t,n	(opposite of) tendency of $< \tilde{s_v}^2 >$
$- < w > \frac{\partial}{\partial z} < \tilde{s_v}^2 >$	BU_SV2	RES ADV	z,t,n	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{s_v}^2 >$	BU_SV2	RES FORC	z,t,n	advection by large-scale W forcing
$- < 2\tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < s_v >$	BU_SV2	RES DP	z,t,n	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{s_v}^2 >$	BU_SV2	RES TR	z,t,n	resolved transport of resolved variance
$- < 2\tilde{s_v}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha s'_v} >$	BU_SV2	RES SBGT	z,t,n	sink due to subgrid turbulence
numerical diffusion of $< \tilde{s_v}^2 >$	BU_SV2	RES NUMD	z,t,n	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{s_v}^2 >$	BU_SV2	RES RELA	z,t,n	sponge layer relaxation
2way nesting of $< \tilde{s_v}^2 >$	BU_SV2	RES NEST	z,t,n	average from smaller nested models
miscellaneous	BU_SV2	RES MISC	z,t,n	chemistry, ...
residual of budget of $< \tilde{s_v}^2 >$	BU_SV2	RES RESI	z,t,n	must be zero
$-\frac{\partial}{\partial t} < \overline{s'^2_v} >$ (neglected in turb. scheme)	BU_SV2	NSG TEND	z,t,n	(neglected) opposite of tendency of $< \overline{s'^2_v} >$
$- < w > \frac{\partial}{\partial z} < \overline{s'^2_v} >$	BU_SV2	NSG ADVM	z,t,n	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{s'^2_v} >$	BU_SV2	NSG ADVR	z,t,n	(neglected) advection by resolved flow

4.5 An illustrative example of a MESONH simulation

We still refer to the example for PREP_IDEAL_CASE for the control of the environment variables. We will still use the rc files to enter the input parameters for the different procedures: prepmodelrc, tosupcrc and the file including the F90 namelist is called EXSEG$n.nam ($n = 1 in this example).

- The final rc file are given below:

  - FILE **prepmodelrc**

```
#! /bin/sh
# default input variables for prepmodel
#-------------------------------------------------------------------------------------
                    #                comfort variables
#OUTSCRIPT=         # instead of outprepmodel, outprepideal, outprep_pgd,
                    # outprepreal, outprepspawn, outprepdiag
#SLEEPSECONDS=      # instead of 10
EDITOR=vi
#
# specific variables
#      Control tools commands (in prepmodel global var) yes or no (default variable)
TOOLSCONTROL=no
#      tori hpce tora (fuji until 20070630) lx____ (32bit)
#      a must have
OUTDEST=
#
#  Automatic submit by tosupc for part one, tosupcrc with TIME MEM NBP TPN has to be OK
LSOUMISAUTO=T
                    # options for treatment of all FM files (and post-treatment)
                    # 'lfiz' 'lfiz conv2dia' 'fm unlfiz' 'all' 'fmmore'
OUTFILE_TOOLS=''
                    # SUBMIT_NEXTJOBS = 0 or filenames to submit super_calculator
                    # at the end of this job
                    # tosupcrc must be present on the same directory
                    # 0=no job
                    # one or several scripts between commas, which will
                    # be submitted one by one by tosupc
```

```
                    # the pathnames are relative to the current $SIMUL directory
                    # or absolute
SUBMIT_NEXTJOBS=0
                    # LOCAL_NEXTJOBS = 0 or filenames to execute workstation jobs
                    # at the end of this job
                    # list of jobs executed on local workstation at the end of this job;
                    # second argument = list of <sh command arguments> which will run
                    # on your workstation with "at" procedure
                    # the pathnames are relative to the current $SIMUL directory or abso
                    # 0=no job
                    # LOCAL_NEXTJOBS = "dirlocal trace1"
                    # the workstation job will execute : cd dirlocal ; sh trace1
LOCAL_NEXTJOBS=0
                    # name of your personal makefile
                    # if you don't have one, general makefile will be executed
                    # on the remote machine, according its OS
MKFNAME=make_mnh
                    # user binary library (0 if none, will be searched from your $HOME)
BIBUSER=0
                    # reference binary library
BIBMASTER=$DEFBIBMASTER
                    # reference bugfix binary library
BIBBUGFIX=$DEFBIBBUGFIX
                    # run with debug options or not (run/debug)
DEBUG=run
                    # loading options added to the default ones :-M|-f|-l perf...
LOAD_OPT=''
        ################ NEW PROGRAM ZOOM_PGD since masdev4_7 ##################
        # are also available PREP_IDEAL_CASE/MODEL/PREP_PGD/DIAG/SPAWNING/PREP_REAL_CAS
        #                    PREP_NEST_PGD
        # other names are possible, in this case do not forget to fill in NAMELISTFILE=
MAINPROG=0
                    # special filename for input namelists
                    # for ZOOM_PGD default namelist is PRE_ZOOM
NAMELISTFILE=default
                    # special list of variables getting from $NAMELISTFILE
```

```
LISTGET=default
                  # ---- input files ---------
                  # location of the input FM files on execution machine
                  # or storage machine ($HOME/$$INDIR) -supc/archiv
INHOST=supc
                  # to get the input FM files, directory name starting
                  # at $HOME if it begins with home/
                  # at $workdir if it begins with work/
                  # indicate one or several directories between double quotes
INDIR="work/INOUTFILES"
                  # user name on which the get will be executed
                  # 0=$LOGNAME
INLOGIN=0
                  # Overview of archiv case:
#INHOST=archiv
#INDIR="../../mxxx/mxxx007/CHEMIN ../mgrp999/NEXTDOOR"
#INLOGIN=0
                  # OR
#INHOST=archiv
#INDIR=NEXTDOOR
#INLOGIN=mrgp999
                  # ---- output files ---------
                  # storage of the output FM files on execution machine
                  # or storage machine -supc/archiv/$RMACH/aerosv2......
OUTHOST=supc
                  # to put the output FM files directory name starting
                  # at $HOME if it begins with home/
                  # at $workdir if it begins with work/
OUTDIR=work/INOUTFILES
#
# global variables used in prepmodel
#
#DEBUGSCRIPT=                          ;# ON /OFF
#ENVIRONMENT=                          ;# SILENTINTERACTIVE/INTERACTIVE
```

- FILE **tosupcrc**

```
#! /bin/sh
# default input variables for tosupc (default values for the NQS parameters)
#  N. Asencio 13/12/94
#------------------------------------------------------------------------
# specific variables
#
                  # filename which contains the script to be run on remote-host
JOBFILE=${1:-0}
                  # jobname
JOBNAME=${2:-$(basename $JOBFILE)}

#################### tosupc will not work unless you correctly fill TIME MEM TPN NBP ###
                  # time in seconds for J2&J3 jobs
TIME=
                  # memory ex: =2000 =2Gb =128Mb (tori: less than 128Gb per node)
                  # be careful : reduce the memory for multi-tasks jobs
MEM=
                  ### tasks per node: only for prep_ideal_case, run or diag
                  ### number of used CPUs from 1 to 8
  ### The most important for parallel execution!
TPN=1
                  # mono or multi-node on NEC (1 to 4) and on IBM
NBP=1
#################### tosupc will not work unless you correctly fill TIME MEM TPN NBP ###

                  # Sending a mail abort + beg and/or at the end job (begend/beg/end/no
                  # default is abort, =no is no mail at all
MAIL=

#
# global variables used in tosupc
#
#DEBUGSCRIPT=                              ; #ON /OFF
#ENVIRONMENT=SILENTINTERACTIVE             ; #SILENTINTERACTIVE/BATCH/INTERACTIVE
```

- Now, you may create a file which contains the following namelists:

  - FILE **EXSEG1.nam** for the previous example of prep_ideal_case

```
&NAM_LUNITn  CINIFILE = "HYD2D" /
&NAM_CONFn  LUSERV = T /
&NAM_DYNn  XTSTEP = 60., LITRADJ= T,
           LHORELAX_UVWTH = T, LVE_RELAX = T, NRIMX = 5, NRIMY = 3,
           XRIMKMAX = .00166,  XT4DIFF = 1500. /
&NAM_PARAMn  CTURB = "TKEL", CRAD = "NONE", CCLOUD = "NONE" /
&NAM_TURBn XIMPL = 1., CTURBLEN = "BL89", CTURBDIM = "1DIM", LTURB_DIAG = T,
           LTURB_FLX = T /
&NAM_LBCn CLBCX = 2*"OPEN", CLBCY = 2*"CYCL", XCPHASE = 20. /
&NAM_CONF CCONF = "START", LTHINSHELL = T, L2D = T, LFLAT = F, NMODEL = 1,
          CEQNSYS="DUR", NVERB = 1, CEXP = "EXPER", CSEG = "HYD2D" /
&NAM_DYN  XSEGLEN =20000., XASSELIN = 0.2, LCORIO = F, XALKTOP = 0.005,
          XALZBOT = 12570., LNUMDIFF =.T. /
&NAM_FMOUT XFMOUT(1,1) = 10000., XFMOUT(1,2) = 20000. /
&NAM_BLANK /
```

- FILE **EXSEG1.nam** for a real case

```
&NAM_LUNITn  CINIFILE = "16J36.1.12B18.001",
              CCPLFILE(1) = "16JAN_06_MNH" /
&NAM_CONFn  LUSERV = T, LUSERC = T, LUSERR = T ,
             LUSERI = T, LUSERS = T, LUSERG =T, LUSECI= T /
&NAM_DYNn  XTSTEP = 75.,  CPRESOPT = "RICHA", NITR = 8,
            LHORELAX_UVWTH = T, LHORELAX_RV = T, LVE_RELAX = T,
            NRIMX = 5, NRIMY = 5, XRIMKMAX = 0.0083, XT4DIFF = 5000. /
&NAM_ADVn  CMET_ADV_SCHEME = "FCT2ND", CSV_ADV_SCHEME = "FCT2ND" /
&NAM_PARAMn  CCLOUD = "ICE3",
              CTURB = "TKEL", CRAD = "ECMWF"
              CGROUND = "ISBA", CDCONV = "KAFR" /
&NAM_PARAM_RADn  XDTRAD = 300., XDTRAD_CLONLY = 150.,
                  NRAD_COLNBR = 400 CAER='TANR' CLW="MORC"/
&NAM_PARAM_CONVECTn XDTCONV = 300., NICE = 1 LREFRESH_ALL = T,
                     LDEEP = T, LSHAL = T, LDOWN = T /
&NAM_PARAM_GROUNDN CROUGH='Z01D' /
&NAM_LBCn  CLBCX = 2*"OPEN", CLBCY = 2*"OPEN" /
&NAM_TURBn  CTURBLEN = "BL89", CTURBDIM = "1DIM", LSUBG_COND=.T., LSUBG_AUCV=T
           LTURB_DIAG=.FALSE.,  LTURB_FLX=.FALSE. LSIG_CONV=T LSIGMAS=F /
```

```
&NAM_CONF  CCONF = "RESTA", NVERB=2,
            NMODEL = 2,
            CEXP = "16J36", CSEG = "12B18" /
&NAM_DYN  XSEGLEN = 800., LCORIO = T, LNUMDIFF = T,
           LSTEADYLS = F, XALKTOP = 0.001, XALZBOT = 14500. /
&NAM_NESTING NDAD(2) = 1, NDTRATIO(2) = 3, XWAY(2) = 2. /
&NAM_FMOUT XFMOUT(1,1) = 10800., XFMOUT(1,2) = 21600. ,
           XFMOUT(2,1) = 10800., XFMOUT(2,2) = 21600. /

&NAM_ISBAn CRUNOFF = "WSAT", CSCOND  = "NP89", CALBEDO = "DRY",
       CC1DRY     = 'DEF', CSOILFRZ   = 'DEF', CDIFSFCOND = 'DEF',
       CSNOWRES   = 'DEF', CROUGH = 'Z04D'/
&NAM_SEAFLUXn CSEA_ALB="UNIF", CSEA_FLUX="DIRECT"/
&NAM_DIAG_SURFn /
&NAM_DIAG_ISBAn /
&NAM_DIAG_SURF_ATMn /
```

The five latest namelists are for the externalised surface.

- To execute the preparative job on the "local_host", enter

  `prepmodel`

  Then answer to the questions asked by the procedure (. for the simulation directory if the environment variable `SIMUL` is not initialized).

  When the job is completed on the "remote_host", an execution report is send back to the "local_host". The model output files are disposed where stated in the prepmodelrc file (in $workdir /INOUTFILES for this example). It should be noted that it is still the case, if the outprepmodel file is executed in an interactive session on the "remote_host" (instead of a batch session).

Chapter 5

Initialization of MESO-NH for real cases

5.1 Overview of the initialization sequences

5.1.1 Initialization for a one-model run, from operationnel models

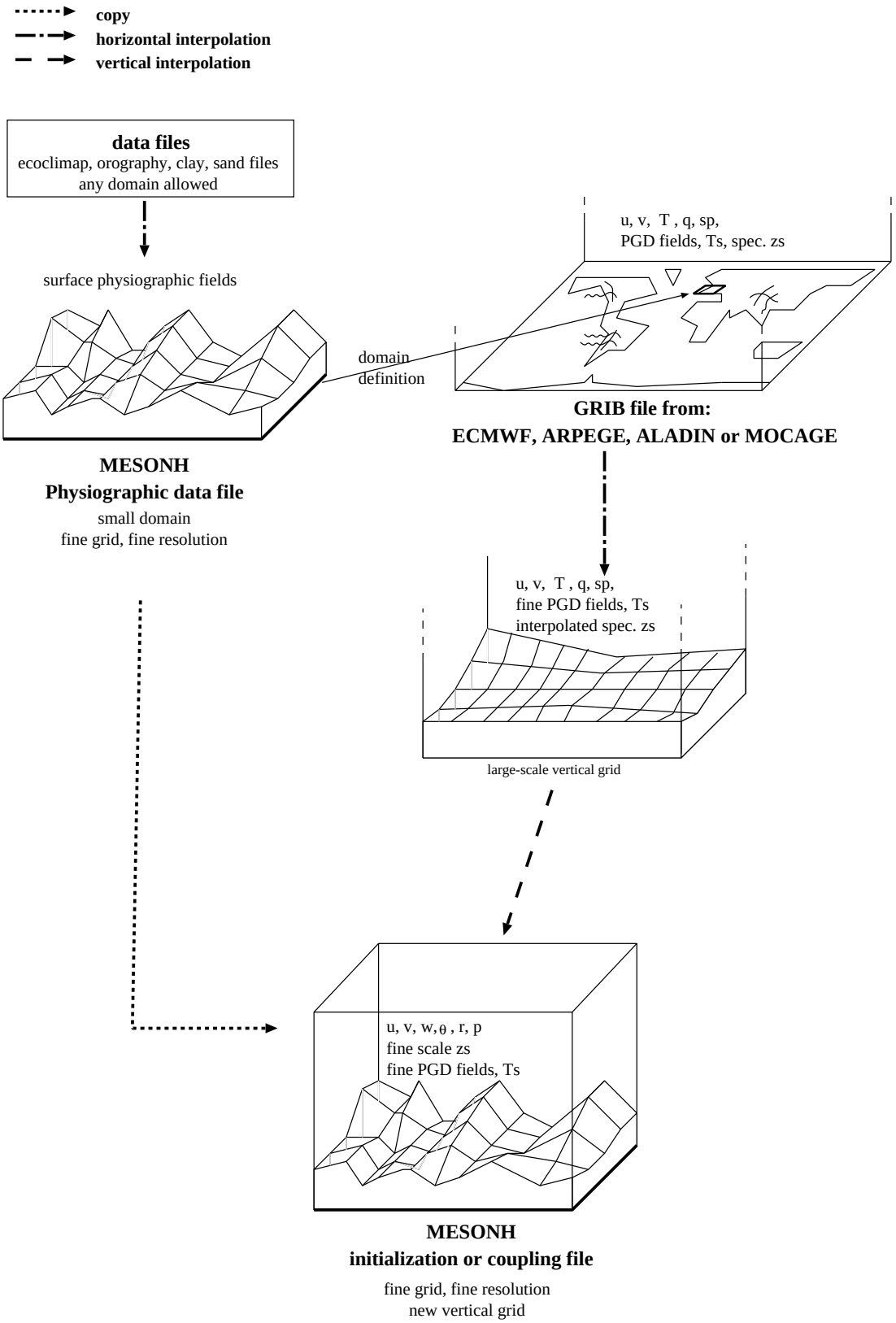
This type of model simulation uses one initial file (with all the fields at beginning time of the simulation), and a certain number of coupling files to prescribe the lateral boundary conditions to the model at further time steps. The coupling files are of the same nature than the initial file. Here is presented the way to produce these initial and coupling files.

This sequence for real cases contains three procedures or programs (figures 5.1 and 5.2). In the following, a fortran program is written in CAPITAL letters, and is run with the MESO-NH procedure **prepmodel**, while an unix procedure is written in lower case letters.

1. **PREP_PGD**: this MESO-NH program computes the physiographic data file (called PGD file below). At this step, you choose the projection, resolution and horizontal domain. The PGD file contains all the physiographic data necessary to run the MESO-NH model with interactive surface schemes for vegetation and town.

2. **extractecmwf** or **extractarpege**: it extracts the surface and altitude fields for one date, respectively for ECMWF archive (ECMWF forecast model) or METEO-FRANCE operational archive (ARPEGE and ALADIN models). In both cases, the fields are written in a GRIB format file, on the gaussian grid. The extraction must be done separately for each date and time (for the initial file and each of the coupling file).

3. **PREP_REAL_CASE**: this MESO-NH program performs the change of orography and vertical grid and writes the MESO-NH file which will be used either for the beginning of the simulation or for coupling.



sp: surface pressure, spec zs: spectral orography (large-scale orography),
Ts: (temperature of surface) stands for all other surface variables: temperatures, water and snow contents,
PGD fields: stands for cover fractions, orographic, vegetation and soil physiographic fields

Figure 5.1: schematic view of the interactions between the different files during the initialization sequence of a real case simulation from a global file of ECMWF or METEO-FRANCE archive.

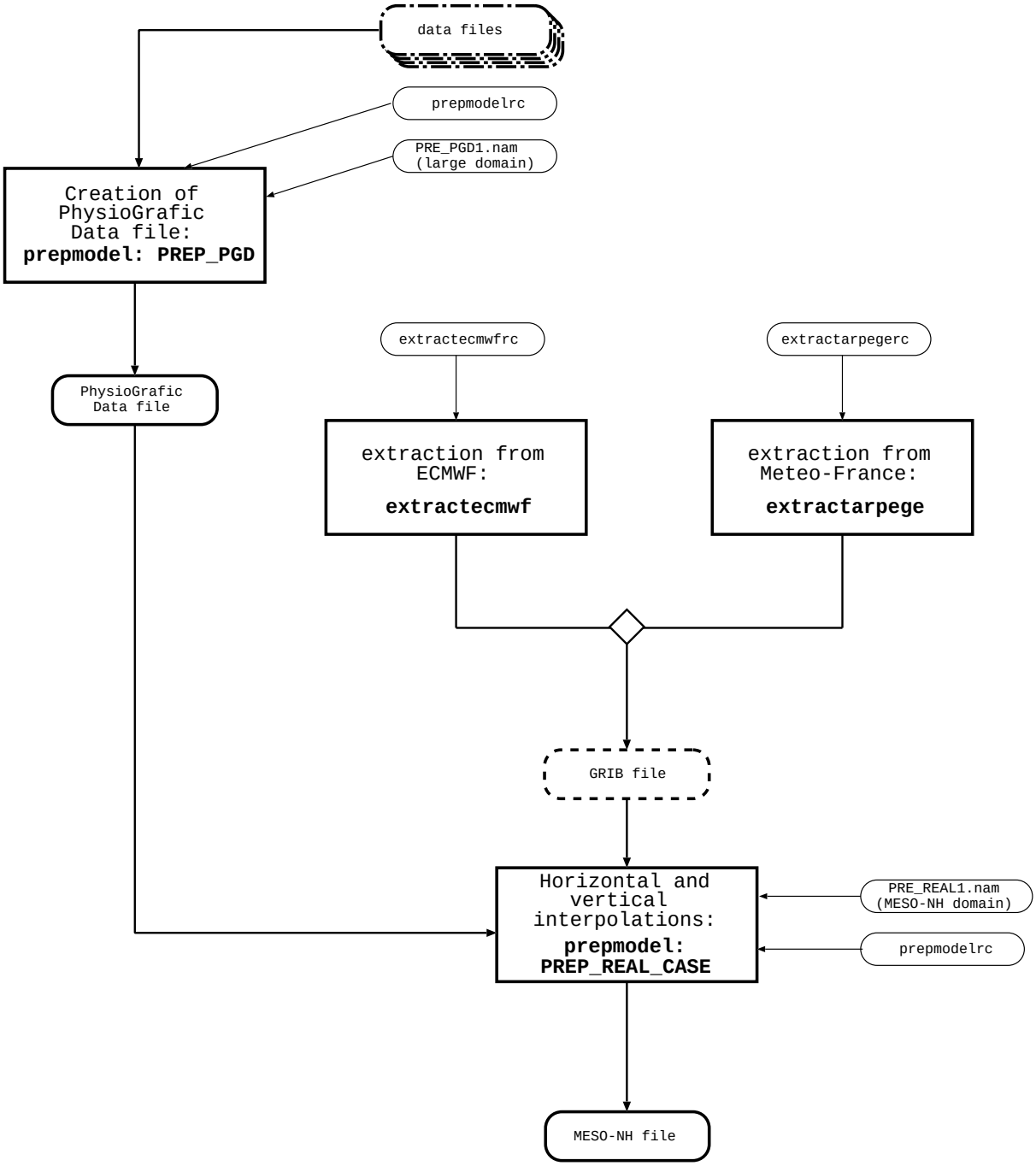


Figure 5.2: initialization sequence of a real case simulation from a global file of ECMWF or METEO-FRANCE archive.

5.1.2 Initialization for a one-model run, from a MESO-NH file

In this case, the initial and coupling files are interpolated from a previous MESONH run (at coarser resolution). It is again necessary to produce separately one initial file and the coupling files.

In this case, the sequence uses (figures (5.3 and 5.4)):

1. **PREP_PGD**
2. **SPAWNING**, this program performs the horizontal interpolations from a MESO-NH file into another MESO-NH file, with a finer resolution and smaller domain.
3. **PREP_REAL_CASE**

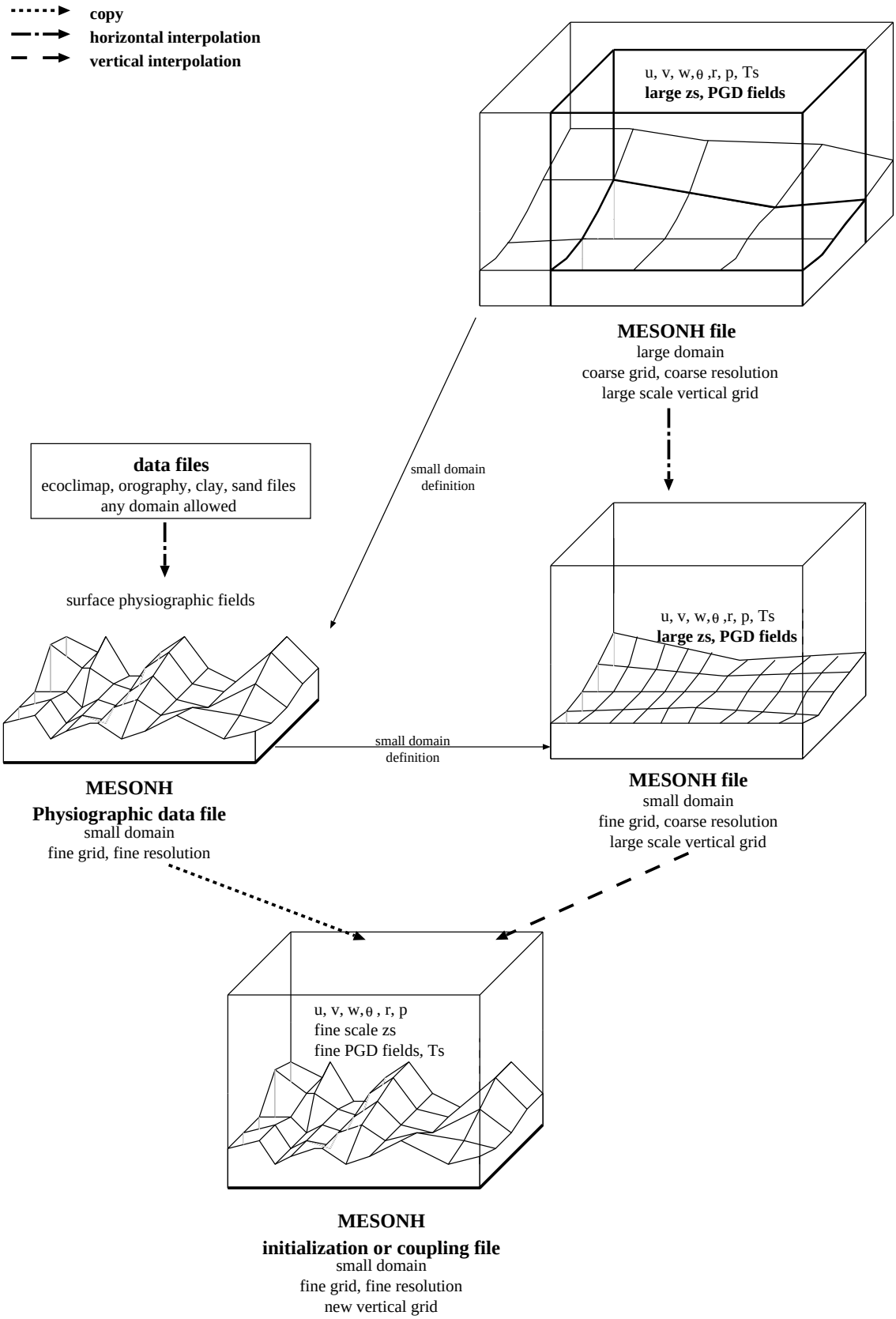
5.1.3 Initialization for a nesting run

Two or more MESO-NH models will be run interactively. The first model, the one with the coarser resolution and containing the others, needs again one initial file and some coupling files (for lateral boundary conditions).

However, the nested models are contained in another MESO-NH model. This allows to give them their lateral boundary conditions directly interpolated from the model which contains it, at all time-steps. Therefore, no coupling file is necessary for those models. The initial file must still be computed before the run (the interpolation program for all the model fields is not yet included in the model program itself).

The user can choose the date of the nested model start as he wants (it is not necessary the same than the coarser model). The only obligation is to start a nested model only if all the coarser models containing it have already started, or are starting.

The initialisation sequence is a merging of the two previous ones (initialisation and coupling files of the first model ; initialisation files from MESO-NH model for the nested files). However, there is a **major change** at the beginning of the sequence: **all the PGD files for all models must be computed before** the **PREP_REAL_CASE** program (and as a consequence before the first model run). Theses PGD files are then checked, and conformity between them for gridnesting is imposed - the orography of one model in every grid mesh is set equal to the average of any of its nested model orography on the same area.



large zs: horizontally interpolated orography (large-scale orography),
Ts: (temperature of surface) stands for all other surface variables: temperatures, water and snow contents,
PGD fields: stands for cover fractions, orographic, vegetation and soil physiographic fields and prognostic fields

Figure 5.3: schematic view of the interactions between the different files during the initialization sequence of a real case simulation from a MESO-NH file.

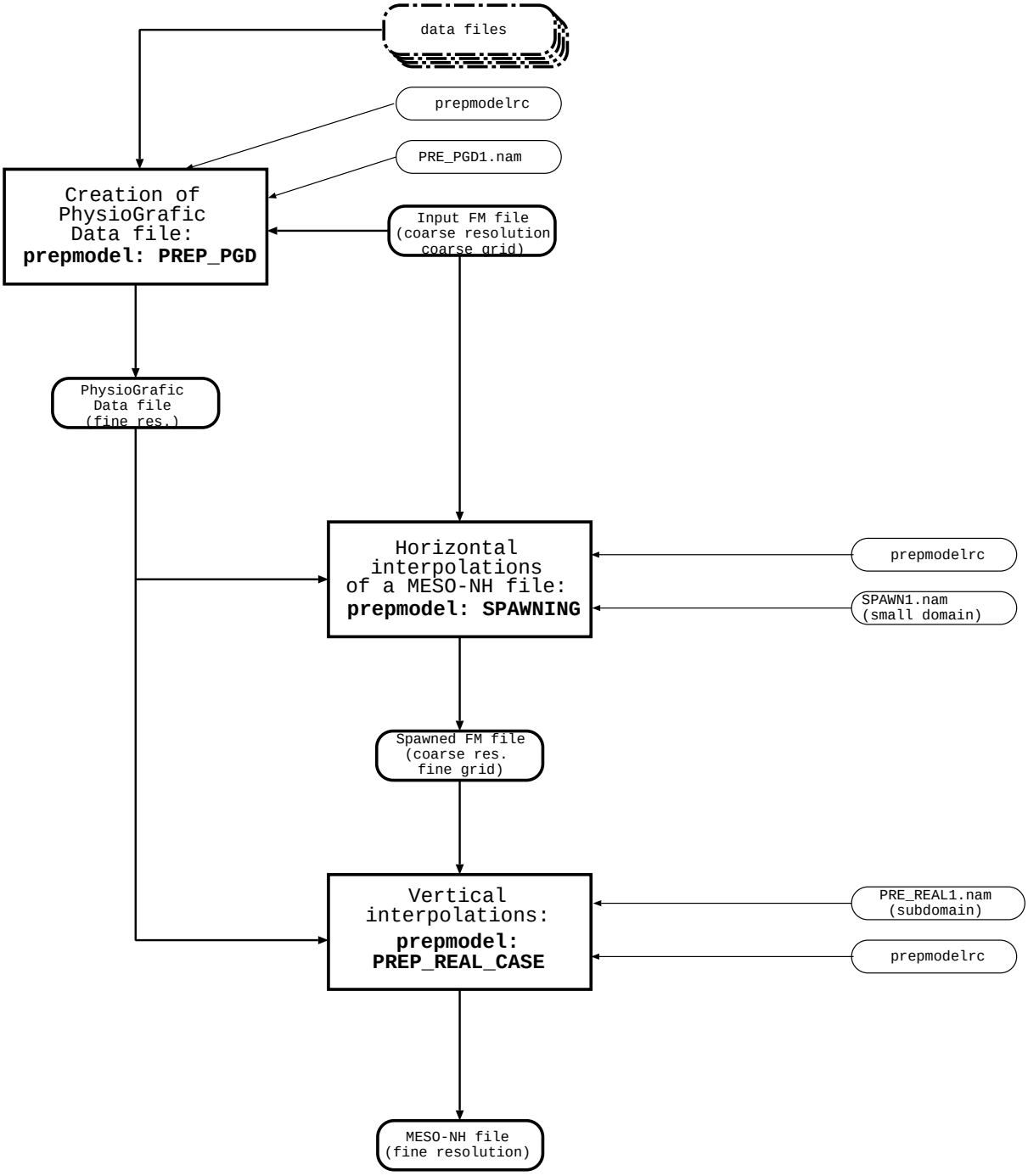


Figure 5.4: initialization sequence of a real case simulation from a MESO-NH file.

- The following initialisation and gridnesting sequence is shown here for three models, model 2 included in model 1, and model 3 included in model 2 (figure (5.5)):

  1. **PREP_PGD**: this program is run as many time as the number of models:
     - one physiographic data file for the model 1 (definition of projection, resolution, domain)
     - one physiographic data file for the model 2 (same projection, definition of resolution, domain)
     - one physiographic data file for the model 3 (same projection, definition of resolution, domain)
  2. **PREP_NEST_PGD**: this program checks all the three PGD files at the same time, and imposes the conformity between them.
  3. **extractecmwf** or **extractarpege**: it extracts the surface and altitude fields for one date, for model 1. The extraction must be done separately for each date and time (for the initial file and each of the coupling file of model 1).
  4. **PREP_REAL_CASE**: this program is running several times, for the initial file and the coupling files of model 1.
  5. **MESONH**: this step is **optional**. If you do not wish to start all the models at the same time, you can decide to run the model 1 before the model 2 starts.
  6. **ZOOM_PGD**: this step is **optional**. If you want to start the model 2 on a smaller domain than the one of the PGD file defined at steps 1 and 2 for the model 2, you must use this program.
  7. **SPAWNING**: when you want to start the model 2, you must use this program to compute the horizontal interpolations from the model 1 to the model 2. It is used only once for the initialisation of model 2.
  8. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 2. **Do not change the vertical grid**.
  9. **MESONH**: again, this step is **optional**. If you do not wish to start model 3 at the same time as model 2, you can decide to run the models 1 and 2 alone before.
  10. **ZOOM_PGD**: again, this step is **optional**. If you want to start the model 3 on a smaller domain than the one of the PGD file defined at steps 1 and 2 for the model 3, you must use this program.
  11. **SPAWNING**: when you want to start the model 3, you must use this program to compute the horizontal interpolations from the model 2 to the model 3. It is used only once for the initialisation of model 3.

12. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 3. **Do not change the vertical grid**.
13. **MESONH**: here is now your complete nested run.

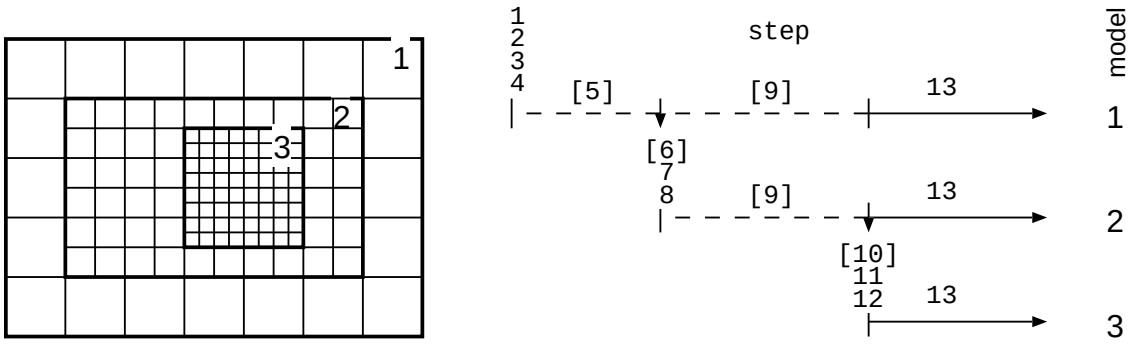


Figure 5.5: Exemple of a grid-nesting simulation with 3 nested models

- The following initialisation and gridnesting sequence is shown here for three models, model 2 included in model 1, and model 3 included in model 1. Model 3 has the same resolution as model 2 and is started after model 2 to follow atmospheric system (figure (5.6)):

  1. **PREP_PGD**:
     - one physiographic data file for the model 1 (definition of projection, resolution, domain)
     - one physiographic data file for the models 2 and 3 (same projection, definition of resolution, domain)
  2. **PREP_NEST_PGD**: this program checks all the two PGD files at the same time, and imposes the conformity between them.
  3. **extractecmwf** or **extractarpege**: it extracts the surface and altitude fields for one date, for model 1. The extraction must be done separately for each date and time (for the initial file and each of the coupling file of model 1).
  4. **PREP_REAL_CASE**: this program is run several times, for the initial file and the coupling files of model 1.
  5. **MESONH**: this step is **optional**. If you do not wish to start all the models at the same time, you can decide to run the model 1 before the model 2 starts.
  6. **ZOOM_PGD**: Since the second PGD file was done for models 2 and 3, you have to zoom it on the domain of model 2 with this program.

7. **SPAWNING**: when you want to start the model 2, you must use this program to compute the horizontal interpolations from the model 1 to the model 2. It is used only once for the initialisation of model 2.
8. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 2. **Do not change the vertical grid**.
9. **MESONH**: here is your complete nested run with model 1 and model 2.
10. **ZOOM_PGD**: Since the second PGD file was done for the models 2 and 3, you have to zoom it on the domain of model 3 with this program. The domain of model 3 has a common zone with the one of model 2.
11. **SPAWNING**: when you want to start the model 3, you must use this program to compute the horizontal interpolations from the model 1 and to use the fields of model 2 in the common domain. It is used only once for the initialisation of model 3.
12. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 3. **Do not change the vertical grid**.
13. **MESONH**: here is the nested run with model 1 and model 3.

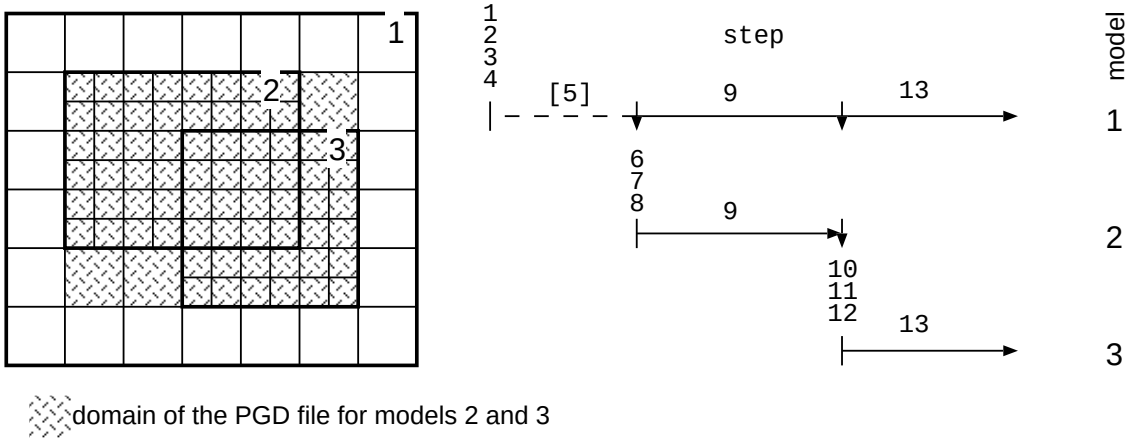


Figure 5.6: Exemple of a grid-nesting simulation with 3 nested models (the domain of the 2 finest models has the same resolution and a common zone to follow atmospheric system).

5.2 Creation of MESO-NH physiographic data file: PREP_PGD

The physiographic fields are averaged or interpolated on the MESO-NH physiographic grid by the program **PREP_PGD**. They are stored in a FM file, called PGD file, but with fewer elements than a MESO-NH file. With the physiographic 2D fields, the geographic and grid data are written in this file. Note that a MESO-NH simulation runs on the grid defined here in **PREP_PGD**.

5.2.1 Execution

PREP_PGD is run with the MESO-NH procedure **prepmodel**.

In the file **prepmodelrc**, the input host, directories and login control variables refer to the input data files (and, if requested, to the FM file YINIFILE). The other control variables to initialize specifically in this file are:

- MAINPROG=PREP_PGD
- NAMELISTFILE=default
- LISTGET=default

A namelist file, PRE_PGD1.nam is also needed.

5.2.2 Presentation of input files

For the version 1 of the externalized surface, **PREP_PGD** needs a minimum of four files to build relatively accurate physiographic fields:

- The **Ecoclimap** file describing the type of cover of the surface, at 30" of resolution. The file is "ecolimap_v2".
- A file containing the orography. The resolution of the MESO-NH team file is 30" on the world. This allows to compute the model orography, and some subgrid-scale orographic characteristics. The file is: "gtopo30".
- A file with the clay fraction of the (near-surface) soil. The resolution of the file of the MESO-NH team is 5' on the world. The file is: "clay_fao".
- A file with the sand fraction of the (near-surface) soil. The resolution of the file of the MESO-NH team is 5' on the world. The file is: "sand_fao".

5.2.3 The input PREP_PGD1.nam file

The file **PRE_PGD1.nam** is needed, containing several namelists. The order of namelists is free and unset namelists can be ommited.

1. Namelist NAM_PGDFILE: (contains file names)

Fortran name	Fortran type	default value
CPGDFILE	character (LEN=28)	' '

   - CPGDFILE : name of the output Physiographic Data File

2. Namelists of the externalized surface for **PREP_PGD**

   As indicated above, the further definition of the surface parameters are not done by MESONH itself, but by the externalized surface included in it. So you are invited to refer to the **documentation of the surface**.

   You must fill the following namelists:

   - NAM_PGD_SCHEMES
   - NAM_PGD_GRID
   - NAM_CONF_PROJ
   - NAM_CONF_PROJ_GRID (if you define the grid completely)
   - NAM_INIFILE_CONF_PROJ (if the grid is a subgrid -eventually with higher resolution- of an existing one)
   - NAM_COVER
   - NAM_ZS
   - NAM_ISBA (if you chose to use the ISBA scheme).
   - NAM_CH_EMIS_PGD
   - NAM_DUMMY_PGD

5.2.4 Examples of PRE_PGD1.nam file

A **PREP_PGD** run where you use the data files provided by the MESO-NH team and some files of your own for the dummy fields:

```
&NAM_PGDFILE        CPGDFILE = 'PGDFILE_1' /

&NAM_PGD_GRID       CGRID = 'CONF PROJ ' /
```

```
&NAM_CONF_PROJ      XLAT0 = 45., XLON0=0., XRPK=0.7, XBETA=0. /
&NAM_CONF_PROJ_GRID NIMAX=100, NJMAX=100, XLATCEN=42.5, XLONCEN=2.5, XDX=10000. , XDY=10000. /
&NAM_INIFILE_CONF_PROJ /
&NAM_PGD_SCHEMES CNATURE='ISBA  ', CSEA='SEAFLX', CTOWN='TEB   ', CWATER='WATFLX' /
&NAM_COVER YCOVER='ecoclimap_v2', YFILETYPE='DIRECT' /
&NAM_ZS    YZS='gtopo30', YFILETYPE='DIRECT' /
&NAM_ISBA  YCLAY='clay_fao', YCLAYFILETYPE='DIRECT',
           YSAND='sand_fao', YSANDFILETYPE='DIRECT',
           XUNIF_RUNOFFB=0.5                          /
&NAM_DUMMY_PGD  NDUMMY_NBR         = 6
                CDUMMY_NAME(1)     = 'SST_2001062100'
                CDUMMY_AREA(1)     = 'SEA'
                CDUMMY_ATYPE(1)    = 'ARI'
                CDUMMY_FILE(1)     = 'SSTn2001062100.dat'
                CDUMMY_FILETYPE(1) = 'ASCLLV'
                CDUMMY_NAME(2)     = 'SST_2001062200'
                CDUMMY_AREA(2)     = 'SEA'
                CDUMMY_ATYPE(2)    = 'ARI'
                CDUMMY_FILE(2)     = 'SSTn2001062200.dat'
                CDUMMY_FILETYPE(2) = 'ASCLLV'
                CDUMMY_NAME(3)     = 'SST_2001062300'
                CDUMMY_AREA(3)     = 'SEA'
                CDUMMY_ATYPE(3)    = 'ARI'
                CDUMMY_FILE(3)     = 'SSTn2001062300.dat'
                CDUMMY_FILETYPE(3) = 'ASCLLV'
                CDUMMY_NAME(4)     = 'SST_2001062400'
                CDUMMY_AREA(4)     = 'SEA'
                CDUMMY_ATYPE(4)    = 'ARI'
                CDUMMY_FILE(4)     = 'SSTn2001062400.dat'
                CDUMMY_FILETYPE(4) = 'ASCLLV'
                CDUMMY_NAME(5)     = 'SST_2001062500'
                CDUMMY_AREA(5)     = 'SEA'
                CDUMMY_ATYPE(5)    = 'ARI'
                CDUMMY_FILE(5)     = 'SSTn2001062500.dat'
                CDUMMY_FILETYPE(5) = 'ASCLLV'
                CDUMMY_NAME(6)     = 'SST_2001062600'
```

```
CDUMMY_AREA(6)     = 'SEA'
CDUMMY_ATYPE(6)    = 'ARI'
CDUMMY_FILE(6)     = 'SSTn2001062600.dat'
CDUMMY_FILETYPE(6) = 'ASCLLV'/
```

Another **PREP_PGD** run for the father PGD file, with all the namelist variables :

```
&NAM_PGDFILE CPGDFILE='PGD_AMMA1_10km_m46_b1' /
&NAM_PGD_SCHEMES  CNATURE='ISBA' , CSEA='SEAFLX', CWATER='WATFLX', CTOWN='TEB'/
&NAM_PGD_GRID CGRID='CONF PROJ'/
&NAM_CONF_PROJ   XLAT0=13. ,XLON0=-1.5, XRPK=0.2249510543, XBETA=0 /
&NAM_CONF_PROJ_GRID  XLATCEN=13. , XLONCEN=-1.5 , NIMAX=324 ,NJMAX=240 ,
                     XDX=10000 , XDY=10000/
&NAM_COVER     YCOVER='ecoclimats_v2', YFILETYPE='DIRECT',LRM_TOWN=.FALSE. /
&NAM_ZS        YZS='gtopo30',    YFILETYPE='DIRECT' ,
               COROGTYPE='AVG', XENV=0., NZSFILTER=1 /
&NAM_ISBA      NPATCH=12, CISBA='3-L',  CPHOTO='AGS', NGROUND_LAYER=3,
               YCLAY='clay_fao', YCLAYFILETYPE='DIRECT' ,
               YSAND='sand_fao', YSANDFILETYPE='DIRECT',
               XUNIF_RUNOFFB=0.5 /
&NAM_CH_EMIS_PGD NEMIS_PGD_NBR=0 /
&NAM_DUMMY_PGD  NDUMMY_PGD_NBR=0 /
```

And for the son PGD file :

```
&NAM_PGDFILE CPGDFILE='PGD_AMMA1_5km_m46_b1' /
&NAM_PGD_SCHEMES  CNATURE='ISBA' , CSEA='SEAFLX', CWATER='WATFLX', CTOWN='TEB'/
&NAM_PGD_GRID YINIFILE='PGD_AMMA1_10km_m46_b1',YFILETYPE='MESONH' /
&NAM_INIFILE_CONF_PROJ IXOR=101 ,IYOR=21 ,IXSIZE=180 ,IYSIZE=150 ,
                      IDXRATIO=2, IDYRATIO=2/
&NAM_COVER     YCOVER='ecoclimats_v2', YFILETYPE='DIRECT',LRM_TOWN=.FALSE. /
&NAM_ZS        YZS='gtopo30', YFILETYPE='DIRECT' ,
               COROGTYPE='AVG', XENV=0., NZSFILTER=1 /
&NAM_ISBA      NPATCH=12, CISBA='3-L',  CPHOTO='AGS', NGROUND_LAYER=3,
               YCLAY='clay_fao', YCLAYFILETYPE='DIRECT' ,
               YSAND='sand_fao', YSANDFILETYPE='DIRECT',
               XUNIF_RUNOFFB=0.5 /
&NAM_CH_EMIS_PGD NEMIS_PGD_NBR=0 /
&NAM_DUMMY_PGD  NDUMMY_PGD_NBR=0 /
```

5.3 Modification of PGD files for grid-nesting: PREP_NEST_PGD

In order to run models with the gridnesting technique, a condition on the orography must be satisfied. In the following, if file #2 is completely included in (and therefore in interaction during the run with) file #1, file #2 will be called the SON file, and file #1 the DAD file. In the following, the DAD file number must be smaller than any of its SON number.

The condition on the orography is: "the mean of orography for a SON file in the domain corresponding to the grid mesh of its DAD file, must be equal to the orography of the DAD file in this mesh".

Such a condition is not automatically satisfied when using enhanced orographies. The program **PREP_NEST_PGD** performs post-treatments on the orographies of up to 8 PGD files that will be used to create initialization files for a gridnested run. It modifies the orography of a DAD from the mean of the orography of its (several) SON(s).

PREP_NEST_PGD is run with the MESO-NH procedure **prepmodel**.

In the file **prepmodelrc**, the input host, directories and login control variables refer to the input PGD files. The other control variables to initialize specifically in this file are:

- MAINPROG=PREP_NEST_PGD
- NAMELISTFILE=default
- LISTGET=default

The namelist file PRE_NEST_PGD1.nam contains: (the order of namelists is free and unset namelists can be ommited)

1. Namelists NAM_PGD**N** (where **N** goes from 1 to 8):
   - YPGD**N**: name of the PGD file #**N**
   - IDAD: number of the DAD file of file #**N**. The DAD file number IDAD must be smaller than #**N**.

2. Namelist NAM_NEST_PGD:
   - YNEST: string of 2 characters to be added to the PGD file names to define the corresponding output PGD file names. The input file *YPGD***N** will be modified into file *YPGD***N**.nest*YNEST*

Example of namelist PRE_NEST_PGD1.nam:

```
&NAM_PGD1 YPGD1 = 'PGDFILE_1' /
&NAM_PGD2 YPGD2 = 'PGDFILE_2', IDAD = 1 /
&NAM_PGD3 YPGD3 = 'PGDFILE_3', IDAD = 1 /
&NAM_PGD4 YPGD4 = 'PGDFILE_4', IDAD = 3 /
&NAM_PGD5 YPGD5 = 'PGDFILE_5', IDAD = 2 /
&NAM_PGD6  /
&NAM_PGD7  /
&NAM_PGD8  /
&NAM_NEST_PGD YNEST = 'e1' /
```

5.4 Zoom of a PGD file: ZOOM_PGD

The previous condition on the orography needed when using the gridnesting technique implies that all the PGD files have to be created (with PREP_PGD and PREP_NEST_PGD programs) **before** beginning the run. However, the user is not always sure where (and when) initialize the inner models. To avoid to set exactly the domain of inner models at the PREP_PGD step, one solution is to make PGD file on larger domain and then, zoom it[1] on the part of the domain of interest when knowned with the following program **ZOOM_PGD**. Then the output PGD file is used as PGD file for the interpolations of atmospheric fields with SPAWNING and PREP_REAL_CASE programs.

ZOOM_PGD is run with the MESO-NH procedure **prepmodel**.

In the file **prepmodelrc**, the input host, directories and login control variables refer to the input PGD file. The other control variables to initialize specifically in this file are:

- MAINPROG=ZOOM_PGD
- NAMELISTFILE=PRE_ZOOM
- LISTGET='CPGDFILE'

The namelist file PRE_ZOOM1.nam contains 2 namelists: (the order of namelists is free)

1. Namelist NAM_PGDFILE: (contains file names)

Fortran name	Fortran type	default value
CPGDFILE	character (LEN=28)	'PGDFILE'
YZOOMFILE	character (LEN=28)	none
YZOOMNBR	character (LEN=2)	'00'

   - CPGDFILE : name of the input Physiographic Data File
   - YZOOMFILE : optional name of the zoomned FM-file (output file). If the user does not specify this name, or if YZOOMFILE = CPGDFILE, the code builds the zoomned FM-file name as:

     YZOOMFILE = CPGDFILE.zYZOOMNBR

   - YZOOMNBR : : NumBeR which will be added to CPGDFILE to generate the name of the Zoomned FM-file (string of 2 characters).

2. Namelist NAM_MESONH_DOM: (contains domain definition variables)

[1] This was done during the PREP_IDEAL_CASE or PREP_REAL_CASE step before the masdev4_6 version.

Fortran name	Fortran type	default value	remarks
NIMAX	integer	input PGD domain	
NJMAX	integer	input PGD domain	
NXOR	integer	NUNDEF	if NUNDEF, domain is in the middle of PGD
NYOR	integer	NUNDEF	if NUNDEF, domain is in the middle of PGD

- NIMAX : number of grid points in I direction, according to input file grid, recovered by the new domain. It must only be factor of 2, 3 or 5.
- NJMAX : number of grid points in J direction, according to input file grid, recovered by the new domain. It must only be factor of 2, 3 or 5.
- NXOR : first point I index, left to and out of the new physical domain.
- NYOR : first point J index, under and out of the new physical domain.

Example of namelist PRE_ZOOM1.nam:

```
&NAM_PGDFILE CPGDFILE = 'PGDFILE_1.neste1' ,
             YZOOMNBR = '58'    /
&NAM_MESONH_DOM NIMAX=60, NJMAX=50,
                NXOR=5, NYOR=8      /
```

5.5 Extraction of ECMWF files: extractecmwf

extractecmwf prepares a job which will perform the extraction of meteorological data at ECMWF using MARS (Meteorological Archival and Retrieval System). Fields are on Gaussian grid or on lat-lon regular grid if limited area is requested, in a standard GRIB file. It can be handled with C routines, such as in library 'pbio' (to make the interface with fortran programs). To treat its information, one needs the library 'gribex'.

Remark: The extraction is executed on **ecgate**: you need an account on it to submit the job (either directly during a session on ecgate, or through an ECaccess gateway).

extractecmwf has control variables. These are given in the file **extractecmwfrc** or interactively. They are:

- OUTHOST= output file will be put or transfered onto a storage machine:

  ecfs: ECMWF local storage system

  ftp-gw: transfer protocol through gateway to the Idris storage machine (gaya), user and password have to be specified in the $HOME/.netrc file of ecgate

  ectrans: ECaccess command via Internet for Meteo-France storage machine (cougar).

- OUTLOGIN= user on the $OUTHOST machine to dispose the desarchived GRIB file.

- OUTDIR= name of directory on the storage machine (for ectrans, the directory has to exist).

- CLASS= archive of extraction (MARS parameter)

  OD: Operational Data

  E4: ERA40 reanalysis[2] from 19570901 to 20020831

  ER: ERA15 reanalysis[3] from 19781201 to 19940228

  something else[4]: for example RD for data from Research Department. In this case, the parameters EXPVER and DATEGAP are also read.

- TYPE= type of extraction for altitude and surface fields: XX-XX with XX (MARS parameter) in

  AN: non initialized analysis

  IA: initialized analysis (only available for CLASS=OD until 19960201)

[2]information can be found at http://www.ecmwf.int/research/era/
[3]information can be found at http://www.ecmwf.int/research/era/ERA-15
[4]see the list at http://www.ecmwf.int/services/archive/d/catalog

FG: first guess (short forecast from previous initialized analysis to smooth the fields)

FC: forecast

AN-AN means AN for both altitude and surface fields,
IA-FG means IA for altitude fields and FG for surface ones.
Orography and land-sea mask surface fields are always requested with TYPE=AN.

- DATE= date, written as YYYYMMDD.
- TIME= validity time (HH) for analysis and first guess, base time for forecast.
- STEP= variable used for first guess or forecast files, it corresponds to

  – the length of forecast of the first guess file at verifying time (i.e. forecast began at TIME-STEP). For time in 00, 06, 12 or 18, STEP must be 06, for time in 03, 09, 15 or 21, STEP can be 03 or 09.

  – the length of forecast since TIME for forecast file (see MARS documentation for allowed values: 6-hourly to TIME+120 then 12-hourly to TIME+240 for dates before 14/11/90; 3-hourly to TIME+12, then 6-hourly to TIME+120 and 12-hourly to TIME+240 for dates since 15/11/90).

- AUTO= automatic definition of grid and area (see below): y/n
  Automatic let you get the maximum resolution available in the archive without interpolation.
- AREA= output area definition (MARS parameter): N lat./ W long./ S lat./ E long. or "G" (for global) or "E" (for Europe: 73.5/-27.0/33.0/45.0). Default (for AUTO=y) is global.
- GRID= output grid mesh definition (MARS parameter): 2 values give E-W and N-S increment latitude/longitude grid (minimum mesh is .5/.5) or a single positive integer specifying the number of line of latitude between the Pole and the Equator in Gaussian output. For AUTO=y, the maximum resolution of Gaussian grid is automatically set according to the date.

Both next variables will allow the user to execute a single extractecmwf for several consecutive dates, beginning from $DATE$TIME for TYPE=AN-AN, or from $DATE$TIME$STEP for TYPE=FC-FC.

- NBLOOP= number of more dates after $DATE$TIME (or $DATE$TIME$STEP).
  0 will do an ordinary extractecmwf with one single date.

- LOOPSTEP= interval between each date extracted (HH).

  Both next variables are needed for MARS data other than Operational Data, ERA40 and ERA15 (example for CLASS=RD for data from Research Department).

- EXPVER= experience version (MARS parameter).

- DATEGAP= gap to add to DATE, written as YYMMDD.
  The date DATE+DATEGAP allows to determine the list of parameters to extract according the version of the IFS model (0 means that the model version is the operational one at the date DATE).

- OUTFILE= name of the GRIB output file.
  When several dates are requested, the number of output file is NBLOOP+1. The name is set to
  `ecmwf.CLASS.DATE.TIME` if TYPE=AN-AN,
  `ecmwf.CLASS.TYPE.DATE.TIME.STEP` if TYPE=FC-FC.

- FLAGTO= automatic submit of the script to ecgate (yes/no)

- SUBMIT_NEXTJOBS= name of one or several jobs to submit to ecgate at the end of this job (works only if extractecmwf is executed on ecgate).

5.6 Extraction of Météo–France files: extractarpege

extractarpege performs the extraction of one GRIB file at Météo–France center in Toulouse: with fields on stretched lat, lon grid for ARPEGE products, on Lambert conformal projection for ALADIN forecast, or on regular lat, lon grid for MOCAGE outputs.

Remark: The extraction is executed on the super computer at Météo–France. This allows for the extracted file to be a standard GRIB file. It can be handled with C routines, such as in library 'pbio' (to make the interface with fortran programs). To treat its information, one needs the library 'gribex'.

Two main steps are performed in the **extractarpege** job execution:

1. the post-processing package FULL-POS[5] is first called to extract altitude and surface fields on the model configuration, for ARPEGE and ALADIN products;

   or a post-processing program written especially[6] in the case of MOCAGE outputs, to gather surface and altitude fields on one the 3 Mocage domains;

2. a tool writing the extracted fields into GRIB format is then run (one is used for ARPEGE and MOCAGE products, another for ALADIN forecast).

extractarpege has control variables. These are given in the file **extractarpegerc** or interactively. They are:

- MODEL= name of the model (ARPEGE, ARPEGE-Tropiques, ALADIN-France, ALADIN-Runion or MOCAGE): arpifs/tropic/aladin/reunion/mocage
- TYPE= for ARPEGE and ALADIN:
  for operational archive: type of altitude and surface fields

  AN: non initialized analysis

  IA: initialized analysis

  FG: first guess (short forecast from previous initialized analysis to smooth the fields)

  FC: forecast,

  for other archive: something else.

  for MOCAGE: type of domain (FRA025, EURO11, GLOB22, RSE088): FR/EU/GL/SE
- DATE= date, written as YYYYMMDD.

[5]Details can be found in the FULL-POS users' guide by Ryad El Khatib (CNRM/GMAP).
[6]Details about MOCAGE model can be found from V.-H. Peuch CNRM/GMGEC/ERAM.

- TIME= validity time (HH) for analysis and first guess, base time for forecast.
- STEP= variable used for forecast files, it corresponds to the length of forecast since TIME.

  Both next variables will allow the user to execute a single extractarpege for operational archive for several consecutive dates, beginning from $DATE$TIME for TYPE=AN, or from $DATE$TIME$STEP for TYPE=FC.

- NBLOOP= number of more dates after $DATE$TIME (or $DATE$TIME$STEP).
  0 will do an ordinary extractarpege with one single date.
- LOOPSTEP= interval between each date extracted (HH).

  The five next variables are needed for special archive (different from operational one, but given in ARPEGE or ALADIN format, e.g. some experiment reanalyses).

- INLOGIN= user on the Météo–France storage-file (archiv) machine, where the input file is stored,
- INDIR= directory of the input ARPEGE or ALADIN file:
  starting at $HOME for archiv machine. home/... = starting at $HOME for super computer storage
  work/... = starting at $workdir for super computer storage
- INFILE= name if the input ARPEGE or ALADIN file,
- LMAX= number of vertical levels present in the input file,
- SOIL= name of the soil scheme used for the input file generation (ISBA, OISB, NONE or other).

- OUTFILE= name of the output GRIB file.
  When several dates are requested, the number of output file is NBLOOP+1. The name is set to
  `MODEL.TYPE.DATE.TIME` if TYPE=AN,
  `MODEL.TYPE.DATE.TIME.STEP` if TYPE=FC.
- OUTHOST= output file will be put onto local the super computer or its storage machine: supc/archiv

- OUTDIR= directory of the output GRIB file:
  starting at $HOME for archiv machine,
  home/... = starting at $HOME for super computer storage,
  work/... = starting at $workdir for super computer storage.

If you choose to extract MOCAGE outputs, you may indicate the list of the chemical species you want to retrieve, in a file named `MOC1.nam`. If not present in your $SIMUL directory, the following list of default values (from the RACM scheme) is taken into account (and copied in `$SIMUL/MOC1.nam` for next use):

```
BEGIN_CONVMOC2GRIB
# RACM species without SULF, ORA1, ORA2
 70 # NUMBER OF OPTIONAL GRIB VARIABLES
180 1 O_3
181 1 H_2O_2
182 1 NO
183 1 NO_2
184 1 NO_3
185 1 N_2O_5
186 1 HONO
187 1 HNO_3
188 1 HNO_4
189 1 DMS
190 1 SO_2
191 1 CO
192 1 O(3P)
193 1 O(1D)
194 1 OH
195 1 HO_2
196 1 CH_4
197 1 ETH
198 1 HC3
199 1 HC5
200 1 HC8
201 1 ETE
202 1 OLT
203 1 OLI
204 1 DIEN
205 1 ISO
206 1 API
207 1 LIM
208 1 TOL
209 1 XYL
210 1 CSL
211 1 HCHO
212 1 ALD
213 1 KET
214 1 GLY
215 1 MGLY
216 1 DCB
217 1 MACR
218 1 UDD
219 1 HKET
220 1 ONIT
221 1 PAN
```

```
222 1 TPAN
223 1 OP1
224 1 OP2
225 1 PAA
226 1 MO2
227 1 ETHP
228 1 HC3P
229 1 HC5P
230 1 HC8P
231 1 ETEP
232 1 OLTP
233 1 OLIP
234 1 ISOP
235 1 APIP
236 1 LIMP
237 1 PHO
238 1 ADDT
239 1 ADDX
240 1 ADDC
241 1 TOLP
242 1 XYLP
243 1 CSLP
244 1 ACO3
245 1 TCO3
246 1 KETP
247 1 OLNN
248 1 OLND
249 1 XO2
END_CONVMOC2GRIB
```

The format used in MOC1.nam is fixed, that is:

- a keyword,

- a comment line,

- the number of chemical species in the output Grib file,

- for each output species: Grib code number, fraction of input species, input species name (*e.g.* `242 1 NO 1 NO_2 2 N_2O_5`),

- a keyword for the end.

5.7 Vertical interpolations: PREP_REAL_CASE

PREP_REAL_CASE performs the vertical interpolations from one orography to another. The main hypothesis is that hydrostatism is verified. Therefore, if the input file is a MESO-NH file, there is a small loss of information. It is possible to use this program to modify the vertical grid, without changing the orography.

What's going in and out?

- Input:
  - the prepmodelrc file
  - a file containing the atmospheric 3D and surface 2D variable fields (hereafter called atmospheric file); it can be either
    - an GRIB file obtained from **extractecmwf** or **extractarpege**
    - or a MESO-NH file (obtained with **SPAWNING** for example)
  - a physiographic data file (it can also be a complete MESO-NH file).
  - an optional file containing the chemical species (here after called chemical file); it is used only if the atmospheric file is a GRIB file. It can be either
    - an GRIB file obtained from **extractarpege** (e.g. an file from the Mocage french model)
    - or a MESO-NH file (obtained in a previous simulation for example)
  - the file PRE_REAL1.nam which contains the directives for PREP_REAL_CASE
- Output:
  - the MESO-NH FM-file

5.7.1 The prepmodelrc file

PREP_REAL_CASE is run with the procedure **prepmodel**. The input and output host, directory and login control variables refer to all the input and output files (and not only to input or output FM files). The other control variables to initialize specifically are:

- MAINPROG=PREP_REAL_CASE
- NAMELISTFILE=default
- LISTGET=default
  These variables contain the names of the input atmospheric file and Physiographic data FM-file respectively.

5.7.2 The physiographic data file

This is a FM file, but with fewer elements than a MESO-NH file. It contains the physiographic 2D fields. The geographic and grid data are stored on this file. This file is created by the program PREP_PGD. It is possible to use a complete MESO-NH file, since it also contains the physiographic fields.

If one wants only modify the vertical grid of a MESO-NH file, without any change on the orography, one can specify it both as atmospheric file and physiographic data file.

It contains:

- the definition of the projection, the horizontal domain and the horizontal grid
- the physiographic fields

5.7.3 The atmospheric file

Both GRIB and FM file are self explanatory. The physiographic data stored in it will not be saved on the output MESONH file.

5.7.4 The chemical file (optional)

Both GRIB and FM file are self explanatory. If the atmospheric file is a GRIB file, the chemical species can be read in another file than the atmospheric one.

5.7.5 The file PRE_REAL1.nam

This file contains namelists with the directives to run PREP_REAL_CASE. The namelists contain the names of the files and the definition of the vertical grid. The order of namelist is free and unset namelists can be ommited. The file can also contain a free formatted part after the vertical grid definition namelist, where are given the vertical levels, if this option is chosen.

1. Namelist NAM_FILE_NAMES: (contains file names)

Fortran name	Fortran type	default value
HATMFILE	character (LEN=28)	' '
HATMFILETYPE	character (LEN=6)	'MESONH'
HPGDFILE	character (LEN=28)	' '
HCHEMFILE	character (LEN=28)	' '
HCHEMFILETYPE	character (LEN=6)	'MESONH'
CINIFILE	character (LEN=28)	'INIFILE'

   - HATMFILE : name of the atmospheric file.
   - HATMFILETYPE : type of the atmospheric file ('GRIBEX', 'MESONH')

- HPGDFILE : name of the Physiographic Data File.
- HCHEMFILE : optional name of the file containing the chemical species if they are not in the HATMFILE or if the ones of the HATMFILE are not to be used (only if HATMFILETYPE is 'GRIBEX'). The grids must be the same as the ones of the output file (CINIFILE).
- HCHEMFILETYPE : type of the chemical file ('GRIBEX', 'MESONH')
- CINIFILE : name of the MESO-NH output FM-file, used as initial or coupling file in a MESO-NH simulation

2. Namelist NAM_REAL_CONF: (contains configuration variables)

Fortran name	Fortran type	default value

CEQNSYS	character (LEN=3)	if HATMFILETYPE = 'GRIBEX':
		'DUR'
		if HATMFILETYPE = 'MESONH':
		CEQNSYS value used in input MESONH file

CPRESOPT	character (LEN=5)	'CRESI'
NVERB	integer	1

LSHIFT	logical	if HATMFILETYPE = 'GRIBEX':
		.TRUE.
		if HATMFILETYPE = 'MESONH':
		.FALSE.

- CEQNSYS : EQuatioN SYStem
  - 'LHE': Lipps-HEmler 1982
  - 'MAE': Modified Anelastic Equations
  - 'DUR': following DURran 1990 derivations
- CPRESOPT : option for pressure solver ('RICHA', 'CGRAD', 'CRESI').
- NVERB : verbosity level (error diagnostics are computed if NVERB>4)
- LSHIFT : flag to shift altitudes in boundary layer

3. Namelist NAM_VER_GRID: (contains vertical grid definition)

The use of the THINSHELL approximation is specified in this namelist. There are five ways to compute the vertical grid (the three first ones are as in PREP_IDEAL_CASE):

(a) constant grid mesh: only the number of levels NKMAX and the grid mesh sizes ZDZGRD and ZDZTOP are used. These must be equal. The type of grid YZGRID_TYPE is set to 'FUNCTN'.

(b) two layers are defined, with constant stretching in each of these, the grid mesh sizes being given near the ground and at top of the model. It is possible that the top grid size is never reached, if the number of points is not enough for the prescribed stretchings. The type of grid YZGRID_TYPE is also set to 'FUNCTN'.

(c) the levels are given by the user. The type of grid YZGRID_TYPE is set to 'MANUAL' in the namelist, and only the number of levels NKMAX is also used in it.

(d) only available when the atmospheric input file is a MESONH file. The levels in the output MESONH file are the same as in the input MESONH file. The type of grid YZGRID_TYPE is set to 'SAMEGR' (for "same grid") and NKMAX is not specified.

(e) only available when the atmospheric input file is a MESONH file. The physical levels of the output MESONH file are the same as the lower NKMAX physical levels in the input MESONH file. The type of grid YZGRID_TYPE is set to 'SAMEGR' (for "same grid") and NKMAX is specified.

The variables of this namelist are:

Fortran name	Fortran type	default value
LTHINSHELL	logical	.FALSE.
NKMAX	integer	60 if HATMFILETYPE='GRIBEX'
	integer	same as in input file if HATMFILETYPE='MESONH'
YZGRID_TYPE	character (len=6)	'FUNCTN' if HATMFILETYPE='GRIBEX'
		'SAMEGR' if HATMFILETYPE='MESONH'
ZDZGRD	real	300 m
ZDZTOP	real	300 m
ZZMAX_STRGRD	real	0 m
ZSTRGRD	real	0 %
ZSTRTOP	real	0 %
LSLEVE	logical	FALSE
XLEN1	real	7500.
XLEN2	real	2500.

- LTHINSHELL : switch for the thinshell approximation (logical)
- NKMAX : number of points in z-direction of the required physical domain. The total size of the array written in initial file will be $NKMAX + 2JPVEXT$ ($JPVEXT$ is fixed to 1 for the present version of Meso-NH)
- YZGRID_TYPE : type of vertical grid definition:
  - 'FUNCTN' : the levels are calculated by the program, according to the namelist variables.
  - 'MANUAL' : the levels are written in the free-formatted part after the namelist.

  - 'SAMEGR' : the levels are the same as those in the input file. Only available when atmospheric input file is a MESONH file.

- ZDZGRD : mesh length in z-direction near the ground
- ZDZTOP : mesh length in z-direction near the top of the model
- ZZMAX_STRGRD : Altitude separating the two constant stretching layers
- ZSTRGRD : Constant imposed stretching (in %) in the lower layer (below ZZMAX_STRGRD)
- ZSTRTOP : Constant imposed stretching (in %) in the upper layer (above ZZMAX_STRGRD)
- LSLEVE : flag for Sleve vertical coordinate.
- XLEN1 : decay scale for smooth topography (in meters)
- XLEN2 : decay scale for smale-scale topography deviation (in meters)

4. namelist NAM_HURR_CONF: (defines hurricane filtering and vortex bogussing)

   Each step (hurricane filtering and vortex bogussing) is separately invoked within the PREP_REAL_CASE program:

   i) The hurricane filtering is applied on four input atmospheric Grib fields (HATMFILETYPE='GRIBEX'), when they are in the horizontal grid of the PGD file and in the vertical grid of the Grib file. The input atmospheric Grib fields filtered are the two horizontal components of wind, the absolute temperature and the surface pressure reduced to ground level. Each field is decomposed into three parts: first, the BASic part is computed by the low-pass Barnes filter; then the hurricane (symmetric) disturbance is computed from the remainder disturbance part. The initial fields are then remplaced by their ENVironmental part: total field minus hurricane disturbance part.

   ii) The vortex bogussing consists on a symmetric vortex added to the input atmospheric MesoNH fields (HATMFILETYPE='MESONH'). The tangential wind is computed from an analytical formulation (Holland, 1980). Then, the balanced mass field is deduced from the thermal wind relation. The bogus of the two horizontal components of wind and the potential temperature is added to the initial (filtered) fields.

   Thus, two PREP_REAL_CASE jobs are to be performed. In a mono-model configuration, the first job allows to remove analysed hurricane from the input Grib fields: filtered and interpolated fields are written in a MesoNH file. It is used as input file for the second PREP_REAL_CASE job during which the analytical vortex is added.

For a grid-nesting simulation, the hurricane filtering is first applied for the outer domain (dad model), with the program PREP_REAL_CASE. The filtered fields are then horizontally interpolated for inner domains with the program SPAWNING (see section 5.8). Then, for each inner domain, a vortex bogussing is added with the program PREP_REAL_CASE.

Fortran name	Fortran type	default value
LFILTERING	logical	.FALSE.
CFILTERING	character (LEN=5)	'UVT '
NK	integer	50
XLAMBDA	real	0.2
XLATGUESS	real	XUNDEF
XLONGUESS	real	XUNDEF
XBOXWIND	real	XUNDEF
XRADGUESS	real	XUNDEF
NPHIL	integer	24
NDIAG_FILT	integer	-1
LBOGUSSING	logical	.FALSE.
XLATBOG	real	XUNDEF
XLONBOG	real	XUNDEF
XVTMAXSURF	real	XUNDEF
XRADWINDSURF	real	XUNDEF
CDADATMFILE	character (LEN=28)	' '
CDADBOGFILE	character (LEN=28)	' '

- LFILTERING : to switch on the filtering of the fields (U,V,T,reduced Ps) of the atmospheric file (logical)
- CFILTERING : to choose the list of the fields to be filtered (U,V,T,reduced Ps).
  - 'UVT ' : U,V,T are filtered (default),
  - 'UVTP ' : U,V,T and reduced PS are filtered,
- NK : number of points of the half-width of the window in which the Barnes filter is applied to compute low-pass component of a given field
- XLAMBDA : a coefficient in the exponential weighting function of the Barnes filter
- XLATGUESS : latitude of the guessed position of the cyclone center
- XLONGUESS : longitude of the guessed position of the cyclone center
- XBOXWIND : half-width of the box inside which the dynamical center is searched from the guessed position (km)
- XRADGUESS : guess of the radius of the domain in which the cyclone will be filtered (km)
- NPHIL : number of azimuthal directions used for the cylindrical coordinates

- NDIAG_FILT : allow storage of several components calculated from total fields. **Be careful, the components are on the Grib vertical grid: in diaprog, plot them only on _K_ levels**. Then to visualize all the Grib vertical levels, the number of MesoNH vertical levels must be equal or greater than the number of levels in the input Grib file.

  0 : total (unfiltered) fields: UT15, VT15 for wind components; TEMPTOT, PRESTOT for absolute temperature and surface pressure,

  environmental (filtered) fields (total field minus hurricane disturbance component): UT16, VT16, TEMPENV, PRESENV,

  0,1 : basic fields (low-pass component isolated by the Barnes filter): UT17, VT17, TEMPBAS, PRESBAS,

  0,1,2 : total disturbance tangential wind component (XVTDIS).
- LBOGUSSING : to switch on the addition of the bogus vortex (logical)
- XLATBOG : latitude of the bogussed position of the analytical cyclone center
- XLONBOG : longitude of the bogussed position of the analytical cyclone center
- XVTMAXSURF: maximum tangential wind near the surface or about 500 m altitude (m/s)
- XRADWINDSURF radius of maximum wind near the surface or about 500 m altitude (km)
- CDADATMFILE: if LBOGUSSING=.TRUE. : name of the dad of HATMFILE.
- CDADBOGFILE: if LBOGUSSING=.TRUE. : name of the dad of CINIFILE. Program will check that CDADATMFILE and CDADBOGFILE have the same characteristics, before replacing the dad name of CINIFILE by CDADBOGFILE instead of CDADATMFILE. CDADBOGFILE must exist before running the prep_real_case job.

5. namelist NAM_AERO_CONF: (defines aerosol initialization)

Fortran name	Fortran type	default value
LDUST	logical	.FALSE.
LORILAM	logical	.FALSE.
XINIRADIUSI	real	0.01
XINIRADIUSJ	real	0.5
CRGUNIT	character (LEN=4)	'MASS'
XINISIGI	real	1.60
XINISIGJ	real	1.60
XN0IMIN	real	10.
XN0JMIN	real	1.

- LDUST : to switch on the initialization of dust aerosols.
- LORILAM : to switch on the initialization of chemical aerosols in case of presence of large scale fields in the atmospheric file (from Mocage model).
- XINIRADIUSI: to switch on the initialization of mean radius mode I (aitken mode) of the distribution (in micrometers).
- XINIRADIUSJ: to switch on the initialization of mean radius mode J (accumulation mode) of the distribution (in micrometers).
- CRGUNIT: type of mean radius given in namelist. Default is for a mass spectral distribution; XINIRADIUSI and XINIRADIUSJ has been converted into a mean radius in number. If not default, you need to give the mean radius for a number spectral distribution (no conversion).
- XINISIGI: value of standard deviation for mode I (Aitken mode).
- XINISIGJ: value of standard deviation for mode J (accumulation mode).

6. NAM_BLANK see Perform a MESONH simulation for description

7. Namelists of the externalized surface for **PREP_REAL_CASE**

   The surface initial fields are produced by **externalized surface** facilities. So you are invited to refer to the **documentation of the surface**. For **PREP_REAL_CASE**, you must fill the following namelists:

   - NAM_PREP_SURF_ATM
   - NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
   - NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
   - NAM_PREP_TEB (if you chose to use the TEB urban scheme)
   - NAM_PREP_ISBA (if you chose to use the ISBA scheme)

   CAUTION:

   (a) Note that all namelists can be void, but only if the initial file name for HATMFILE you provide in namelist NAM_FILE_NAMES contains externalized surface fields.

   (b) If the file HATMFILE does not contains externalized surface fields, you must fill at least namelist NAM_PREP_SURF_ATM (if you want to initialize the surface prognostic fields from a input file). You can also define more precisely the surface fields by using the namelists for each scheme.

8. Free formatted part : Vertical grid

   This part is optional in the file, read only if YZGRID_TYPE='MANUAL'. It must begin by the keyword **ZHAT** In this case (NKMAX+1) levels are written in meters in free format after the keyword, from ground level (generally 0) to rigid top level.

9. Second free formatted part related to chemical species: This part is only used if you have previously run **extractarpege** with MOCAGE outputs. This part has to be written at the end of the namelist file. In this case, the list of the MesoNH chemical species, and their corresponding grib code in the Grib file, is specified as follows:

```
MOC2MESONH
transfer mocage/RACM variables (default values)
2 # NUMBER OF OPTIONAL GRIB VARIABLES
(A4,1X,I5)
O3   180
NO2  183
```

   If you only indicate `MOC2MESONH` the list of default values corresponding of the default ones for extractarpege is read in `$MESONH/procedures/PRE_MOC1.nam` and add to your own PRE_REAL1.nam.

Examples of namelist file PRE_REAL1.nam

- Gribex file, levels being calculated, and chemical species

```
&NAM_FILE_NAMES HATMFILE   ='ALT90101500'     , HATMFILETYPE='GRIBEX' ,
                HPGDFILE   ='PGDFILE_10km'    ,
                CINIFILE   ='CPL_example1'    /
&NAM_REAL_CONF CEQNSYS="LHE", NVERB=7 /
&NAM_VER_GRID NKMAX=60, YZGRID_TYPE='FUNCTN', ZDZGRD=50., ZDZTOP=500.,
              ZZMAX_STRGRD=3000., ZSTRGRD=2., ZSTRTOP=6. /
&NAM_BLANK


MOC2MESONH
transfer mocage/RACM variables
2 # NUMBER OF OPTIONAL GRIB VARIABLES
(A4,1X,I5)
O3   180
NO2  183
```

N.B.: the mocage part is written at the end of namelist file.

- MESONH file and levels given manually

```
&NAM_FILE_NAMES HATMFILE   ='POI03.1.DAY01.001'  , HATMFILETYPE='MESONH' ,
                HPGDFILE   ='PGDFILE_10km'       ,
                CINIFILE   ='CPL_example2'       /
&NAM_REAL_CONF NVERB=5 /
&NAM_VER_GRID NKMAX=10, YZGRID_TYPE='MANUAL' /
ZHAT
0.
1050.
2100.
3250.
4300.
5200.
6100.
7000.
8000.
9000.
10000.
```

5.8 Horizontal interpolation from a MESO-NH file: SPAWNING

This program performs the horizontal interpolation from one MESO-NH file into another (respectively file 1 and file 2). The grid of the file 2 must be exactly included in the grid of file 1. The file 2 can be directly used for a model run, but it contains smooth surface fields (expecially the orography). It is possible to run the model with the two files with gridnesting interaction, since a iterative procedure insures the gridnesting condition on the orographies.

The domain of the file 2 can be defined either:

1. by namelist NAM_GRID2_SPA specification.
2. with the domain of another FM file, which grid is coherent with the input file. For example this file can be a PGD file created by **PREP_PGD** with a domain defined from the domain of file 1 and the same type of specifications as those in NAM_GRID2_SPA (see above).

5.8.1 The prepmodelrc file

SPAWNING is run with the procedure **prepmodel**. The input and output host, directory and login control variables refer to all the input and output files. The other control variables to initialize specifically are:

- MAINPROG=SPAWNING
- NAMELISTFILE=default
- LISTGET=default

5.8.2 The input SPAWN1.nam file

The order of namelists is free and unset namelists can be ommited.

1. Namelist NAM_GRID2_SPA: (manual definition of domain)

Fortran name	Fortran type	default value
IXOR	integer	1
IYOR	integer	1
IXSIZE	integer	file 1 domain
IYSIZE	integer	file 1 domain
IDXRATIO	integer	1
IDYRATIO	integer	1
GBAL_ONLY	logical	.FALSE.

   - IXOR: first point I index, according to the file 1 grid, left to and out of the new physical domain.

- IYOR: first point J index, according to the file 1 grid, under and out of the new physical domain.
- IXSIZE: number of grid points in I direction, according to file 1 grid, recovered by the new domain. It must only be factor of 2,3 or 5.
- IYSIZE: number of grid points in J direction, according to file 1 grid, recovered by the new domain. It must only be factor of 2,3 or 5.
- IDXRATIO: resolution factor in I direction between the file 1 grid and the new grid. It must only be factor of 2,3 or 5.
- IDYRATIO: resolution factor in J direction between the file 1 grid and the new grid. It must only be factor of 2,3 or 5.
- GBAL_ONLY: switch to enforce anelastic constraint only. The spawned file have the same caracteristics than the CINIFILE one.

2. Namelist NAM_LUNIT2_SPA: (file names)

Fortran name	Fortran type	default value
CINIFILE	character (len=28)	'INIFILE'
YDOMAIN	character (len=28)	none
YSPAFILE	character (len=28)	none
YSPANBR	character (len=2)	'00'
YDADINIFILE	character (len=28)	"
YDADSPAFILE	character (len=28)	"
YSONFILE	character (len=28)	"

- CINIFILE : name of the initial FM-file 1 (father domain) which will be used to spawn model 2.
- YDOMAIN : name of the file which defines the domain for model 2. If a domain file is provided for YDOMAIN, then all the information of namelist NAM_GRID2_SPA will be ignored.
- YSPAFILE : optional name of the spawned FM-file 2 (output file). If the user does not specify this name, or if YSPAFILE = CINIFILE, the code builds the spawned FM-file name as:

  YSPAFILE = CINIFILE.spaYSPANBR

  or YSPAFILE = CINIFILE.sprYSPANBR if YSONFILE is provided.
- YSPANBR : : NumBeR which will be added to CINIFILE to generate the FM-file name of the SPAwned file (string of 2 characters)
- YDADINIFILE : if GBAL_ONLY=.TRUE. : name of the dad of CINIFILE.

- YDADSPAFILE : if GBAL_ONLY=.TRUE. : name of the dad of YSPAFILE. Program will check that YDADINIFILE and YDADSPAFILE have the same characteristics, before replacing the dad name of YSPAFILE by YDADSPAFILE instead of YDADINIFILE. YDADSPAFILE must exist before running the spawning job.
- YSONFILE : optional name of a spawned FM-file (input file). It must have the same resolution as the spawned FM-file 2 (output file). The fields of YSONFILE will be used at points included in the domain defined by YDOMAIN or NAM_GRID2_SPA, instead of interpolated fields of CINIFILE. This allows to keep finest information when defining a new finest domain to follow atmospheric system.

3. NAM_BLANK see Perform a MESONH simulation for description

4. Namelist NAM_SPAWN_SURF:

Fortran name	Fortran type	default value
LSPAWN_SURF	logical	.TRUE.

- LSPAWN_SURF : : flag to choose to or not to perform the interpolation of all the surface fields (both physiographic and prognostic fields). Note that these interpolations are performed by **externalized surface** facilities. However, no specific namelist is required for this operation.
  - LSPAWN_SURF = .TRUE. : the interpolations are made
  - LSPAWN_SURF = .FALSE. : the interpolations are not made and therefore no surface field will be present in the output spawned file.

5.9 If you want surface fields only: PREP_SURFEX

PREP_SURFEX performs the interpolations of surface fields from one grid to another.

What's going in and out?

- Input:
  - the prepmodelrc file
  - a file containing the surface 2D variable fields (hereafter called input file); it can be either
    - an GRIB file obtained from **extractecmwf** or **extractarpege**
    - a MESO-NH file (obtained with **SPAWNING** for example)
  - a physiographic data file (it can also be a complete MESO-NH file).
  - the file PRE_REAL1.nam which contains the directives for PREP_SURFEX
- Output:
  - the FM-file containing the physigraphic and pronostic surface fields.

5.9.1 The prepmodelrc file

PREP_SURFEX is run with the procedure **prepmodel**. The input and output host, directory and login control variables refer to all the input and output files (and not only to input or output FM files). The other control variables to initialize specifically are:

- MAINPROG=PREP_SURFEX
- NAMELISTFILE=default
- LISTGET=default
  These variables contain the names of the input atmospheric file and Physiographic data FM-file respectively.

5.9.2 The physiographic data file

This is a FM file, but with fewer elements than a MESO-NH file. It contains the physiographic 2D fields. The geographic and grid data are stored on this file. This file is created by the program PREP_PGD. It is possible to use a complete MESO-NH file, since it also contains the physiographic fields.

If one wants only modify the vertical grid of a MESO-NH file, without any change on the orography, one can specify it both as atmospheric file and physiographic data file.

It contains:

- the definition of the projection, the horizontal domain and the horizontal grid
- the physiographic fields

5.9.3 The input file

Both GRIB and FM file are self explanatory. The physiographic data stored in it will not be saved on the output file.

5.9.4 The file PRE_REAL1.nam

This file contains namelists with the directives to run PREP_SURFEX. The namelists contain the names of the files. The order of namelist is free and unset namelists can be ommited.

1. Namelist NAM_FILE_NAMES: (contains file names)

Fortran name	Fortran type	default value
HATMFILE	character (LEN=28)	' '
HATMFILETYPE	character (LEN=6)	'GRIBEX'
HPGDFILE	character (LEN=28)	' '
CINIFILE	character (LEN=28)	'INIFILE'

   - HATMFILE : name of the atmospheric file (up to 28 characters)
   - HATMFILETYPE : type of the atmospheric file ('GRIBEX', 'MESONH')
   - HPGDFILE : name of the Physiographic Data File (up to 28 characters)
   - CINIFILE : name of the MESO-NH output FM-file, used as initial or coupling file in a MESO-NH simulation

2. externalized surface namelists for **PREP_SURFEX**

   The surface initial fields are produced by **externalized surface** facilities. So you are invited to refer to the **documentation of the surface**. For **PREP_SURFEX**, you must fill the following namelists:

   - NAM_PREP_SURF_ATM
   - NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
   - NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
   - NAM_PREP_TEB (if you chose to use the TEB urban scheme)
   - NAM_PREP_ISBA (if you chose to use the ISBA scheme)

CAUTION:

(a) Note that all namelists can be void, but only if the initial file name for HATMFILE you provide in namelist NAM_FILE_NAMES contains externalized surface fields.

(b) If the file HATMFILE does not contains externalized surface fields, you must fill at least namelist NAM_PREP_SURF_ATM (if you want to initialize the surface prognostic fields from a input file). You can also define more precisely the surface fields by using the namelists for each scheme.

5.10 Miscellaneous interactive procedures

A set of procedures runing interactively are available with the MesoNH environment:

5.10.1 Content of a MESO-NH file: fmmore

This procedures lists the fields stored in a FM file (MESONH or PGD file), gives the projection and domain definition, the number of points in each direction, the horizontal grid meshes, the entire vertical grid, and the time informations of the file. The descriptive part of the FM file is also printed (see an example in 8.2.1).

This procedure is used as:

fmmore *MESONH_file_name*

5.10.2 Definition of the vertical grid: vergrid

In order to help the user to define ist vertical grid with the FUNCTN type, this procedure computes the new grid as done in **PREP_IDEAL_CASE** or **PREP_REAL_CASE**.

It uses a namelist file containing only the namelist NAM_VER_GRID containing the variables LTHINSHELL, NKMAX, YZGRID_TYPE, ZDZGRD, ZDZTOP, ZZMAX_STRGRD, ZSTRGRD, ZSTRTOP described above in **PREP_REAL_CASE** presentation.

This procedure is used as:

vergrid *namelist_file_name*

5.10.3 Conversion from spherical coordinates to conformal coordinates: latlon2xy

This procedures computes the position of points given with their latitude and longitude onto the conformal plane (coordinates x,y and index i,j) for a FM file (with its projection, domain and grid).

This procedure is used as:

latlon2xy *MESONH_file_name*

Afterwards, the user follows the indications of the program (he gives interactively the latitudes and longitudes of the points).

5.10.4 Conversion from conformal coordinates to sphericalcoordinates: xy2latlon

This procedures computes the position of points given with coordinates x,y on the conformal plane of a FM file (with its projection, domain and grid), onto the earth (latitude and longitude).

This procedure is used as:

xy2latlon *MESONH_file_name*

Afterwards, the user follows the indications of the program (he gives interactively the x and y coordinates of the points).

Chapter 6

Compute diagnostics after a MESO-NH simulation

6.1 Presentation

After running the model, useful quantities can be diagnosed from prognostic variables contained in the FM output files. It is done by the program **DIAG** which computes diagnostic variables. If you choose diachronic format for output file, you can plot fields with the interactive program `diaprog`[1], in the super-computer or in your workstation.

Available diagnostics are listed in section 6.3. They can be sorted into 5 main categories as seen by the 5 tables. You can add your own diagnostics by modifying the source *write_lfifm1_for_diag.f90* or *write_lfifm1_for_diag_supp.f90*

6.2 Execution

DIAG is run with the Meso–NH procedure **prepmodel** used with the namelist `DIAG1.nam`. An example of the namelist file is given in section 6.2.2. Up to 24 FM files can be treated identically in a single job.

Two main tasks are performed by the procedure **prepmodel**:

1. After the get of the input FM file, diagnostics are computed

   by the routine *write_lfifm1_for_diag*. Diagnostic variables, as well as some prognostic variables, are written in a temporary FM file;

   by call to the physic parametrisations monitor *phys_param1*,

   by the routine *write_lfifm1_for_diag_supp*. Diagnostic variables related to convection, radiation and surface, computed in the physic parametrisations monitor, are written here.

[1] see the graphic documentation by J. Duron on the Meso-NH web site

2. Each temporary FM file can be converted into a diachronic file by the program `conv2dia` (see "TRAITEMENT GRAPHIQUE DES FICHIERS SYNCHRONES produits PAR LE MODÈLE MESONH" J. Duron) called by the script **transfer.x** from **prepmodel**, or kept in FM format and renamed.

   These output files are then stored (in a workspace in the super-computer or in the archive machine) or transferred to your workstation.

6.2.1 The prepmodelrc file

In the file **prepmodelrc**, the input and output host, directories and login control variables refer to the input and output files. The other control variables to initialize specifically in this file are:

- MAINPROG=DIAG
- NAMELISTFILE=default
- LISTGET=default
  this variable contains the names of the input FM files (up to 24).
- OUTHOST=name_workstation (for example)
  this allows future use of `diaprog` on your workstation if you chose to convert into a diachronic file.

6.2.2 The namelist file DIAG1.nam

The `DIAG1.nam` namelist file contains the diagnostics required by the user, the name of the input FM files, the suffix of the output diachronic files, and output file type. The user can reset options for the convective and radiation scheme with NAM_PARAM_CONVECTn and NAM_PARAM_RADn namelist (see chapter 4) The order of namelists is free and unset namelists can be ommited.

1. Namelist NAM_DIAG (controls diagnostic variables)

Fortran name	Fortran type	default value
CISO	character (len=6)	'PREVTK'
LVAR_RS	logical	.TRUE.
all other logicals		.FALSE.
all other character strings		' '
all integers		-1
LDIAG	array of logicals	100*.FALSE.
XDIAG	array of reals	100*XUNDEF

see section 6.3 for meaning

Fortran name	Fortran type	default value
LAIRCRAFT_BALLOON	logical	.FALSE.
NTIME_AIRCRAFT_BALLOON	integer	NUNDEF
XSTEP_AIRCRAFT_BALLOON	real	XUNDEF
XLAT_BALLOON	array(real)	9*XUNDEF
XLON_BALLOON	array(real)	9*XUNDEF
XALT_BALLOON	array(real)	9*XUNDEF

- LAIRCRAFT_BALLOON: flag to compute aircraft and balloon trajectories with stationnary fields. Trajectories will be writen in diachronic file *YINIFILE*BAL
- NTIME_AIRCRAFT_BALLOON: time length of trajectories computation centered on CURrent time (s)
- XSTEP_AIRCRAFT_BALLOON: minimum time step for trajectories computation (s)
- XLAT_BALLOON: initial latitudes of the balloons
- XLON_BALLOON: initial longitudes of the balloons
- XALT_BALLOON: initial altitudes of the balloons (m)

2. Namelist NAM_DIAG_BLANK

Fortran name	Fortran type	default value
XDUMMY_DIAG	array(real)	20* 0.
NDUMMY_DIAG	array(integer)	20* 0
LDUMMY_DIAG	array(logical)	20* TRUE
CDUMMY_DIAG	array(80 characters)	20* ”

identical use than NAM_BLANK (see chapter 4). Add USE MODD_DIAG_BLANK in a diag subroutine to use any of these variables.

3. Namelist NAM_DIAG_FILE

Fortran name	Fortran type	default value
YINIFILE	array of character (len=28)	24*’ ’
YSUFFIX	character (len=5)	’_DIAG’

- YINIFILE: name of the input FM files.
- YSUFFIX: suffix appended to input file name to form output file name.

4. Namelist NAM_STO_FILE (controls trajectories computation)
only read if LTRAJ=.TRUE. in NAM_DIAG

Fortran name	Fortran type	default value
CFILES	array of character (len=28)	100*' '
NSTART_SUPP	array of integer	100*NUNDEF

- CFILES: name of all the input FM files taken into account to compute trajectories. They must be in **inverse** chronological order, and correspond to a reinitialisation of Lagrangian tracers (see chapter 4).
- NSTART_SUPP: extra origins for trajectories computation. In the second example of 6.2.3, the output files will contain the set of variables (X000, Y000, Z000, TH000, RV000) for origin corresponding to the last file (CFILES(6)), and 2 extra sets (X`n`, Y`n`, Z`n`, TH`n`, RV`n`) with `n`=001 for origin corresponding to the CFILES(4), `n`=002 corresponding to CFILES(2). (Note that extra origins are in chronological order).

5. Namelist NAM_PARAM_CONVECTn (options for the convective scheme when convective diagnostics with NCONF_KF)
   see chapter 4 for variables meaning.

6. Namelist NAM_PARAM_RADn (options for the radiations when radiation diagnostics with NRAD_3D)
   see chapter 4 for variables meaning.

7. Namelists of the externalized surface
   Diagnostics of the surface parameters are not computed by MESONH itself, but by the externalized surface included in it. So you are invited to refer to the **documentation of the surface**.

   - namelist NAM_DIAG_SURF_ATMn
   - namelist NAM_DIAG_SURFn
   - namelist NAM_DIAG_ISBAn
   - namelist NAM_DIAG_TEBn
   - namelist NAM_DIAG_CSEAn
   - namelist NAM_DIAG_WATERn

6.2.3 Examples of DIAG1.nam

- Namelist file for 2 files

```
&NAM_DIAG
  LVAR_LS=T,
  NCONV_KF=2,
```

```
  NRAD_3D=1,
  CRAD_SAT='METEOSAT'
  LVAR_MRW=T, LVAR_MRSV=T,
  LMOIST_V=T, LMOIST_E=F,
  LTPZH=T, LVORT=F, LMSLP=F,
  LGEO=T, LAGEO=T, LWIND_ZM=F,
  LTHW=T, LCLD_COV=T,
  LVAR_PR=F, LTOTAL_PR=F, LMEAN_PR=F, XMEAN_PR(1,2)=4. ,
  NCAPE=1,
  LRADAR=T,
  LTRAJ=F /
&NAM_DIAG_BLANK /
&NAM_DIAG_FILE  YINIFILE(1) = "F9801.1.06A12.002" ,
                YINIFILE(2) = "F9802.1.06A12.002" ,
                YSUFFIX = "diag" /
&NAM_DIAG_ISBAn N2M=2, LSURF_BUDGET=T /
```

- Namelist file for 6 files using trajectories computation

```
&NAM_DIAG
  LVAR_PR=T, LTOTAL_PR=T,
  LTPZH=T, LVAR_MRSV=T,
  LTRAJ=T /
&NAM_DIAG_FILE  YSUFFIX='d18-6'
                YINIFILE(1) = "NAPE2.1.APE05.001" ,
                YINIFILE(2) = "NAPE2.1.APE04.001" ,
                YINIFILE(3) = "NAPE2.1.APE03.001" ,
                YINIFILE(4) = "NAPE2.1.APE02.001" ,
                YINIFILE(5) = "NAPE2.1.APE01.001" ,
                YINIFILE(6) = "APE10_ARP19990919.18"
&NAM_STO_FILE   CFILES(1) = "NAPE2.1.APE05.001" ,
                CFILES(2) = "NAPE2.1.APE04.001" ,
                CFILES(3) = "NAPE2.1.APE03.001" ,
                CFILES(4) = "NAPE2.1.APE02.001" ,
                CFILES(5) = "NAPE2.1.APE01.001" ,
                CFILES(6) = "APE10_ARP19990919.18" ,
                NSTART_SUPP(1)= 4 ,
                NSTART_SUPP(2)= 2 /
```

6.3 Variables available in the output diachronic file

Atmospheric variables (prognostic or diagnostic)	
`Variables` [dim] meaning (unit)	in `DIAG1.nam`
`RHODREF` : [3D] Dry density for reference state with orography (kg/m^3)	
`THVREF` : [3D] Thetav for reference state with orography (K)	
Model fields:	
`PABSM` : [3D] pression (Pa)	`CISO='PR','PREVTK'`
`THM` : [3D] potential temperature (K)	`CISO='TK','PRTK'`
`UM, VM, WM` : [3D] wind components (m/s)	`LVAR_RS`
`RVM` : [3D] water vapor mixing ratio (kg/kg)	''
`LSUM, LSVM, LSWM, LSTHM, LSRVM` : [3D] Large Scale variables	`LVAR_LS`
All wind components and wind derivate components	`LWIND_ZM`
suffixed by `_ZM` for zonal and meridien components	
FoRCing variables at instant n: (`LFORCING` in YINIFILE.des)	
`UFRCn, VFRCn, WFRCn`: [1D] (m/s)	`LVAR_FRC`
`THFRCn, RVFRCn, TENDTHFRCn, TENDRVFRCn` : [1D] (/s)	''
`GXTHFRCn, GYRVFRCn`: [1D] ()	''
`PGROUNDFRCn`: [0D] ()	''
Convective scheme:	
`CAPE` : [2D] Convective Available Potentiel Energy (J/kg)	`NCONV_KF=0,1,2`
`CLTOPCONV, CLBASCONV` : [2D] top and base of conv. clouds (km)	''
`DTHCONV` : [3D] Convective tendency for potential temperature (K/s)	`NCONV_KF=1,2`
`DRVCONV, DRCCONV, DRICONV` : '' for vapor, cloud, ice (/s)	''
`DSVCONVnnn` : '' for scalar variable `n` (/s)	''
`UMFCONV, DMFCONV` : [3D] Updraft and Downdraft Mass Flux (m^2 kg/s)	`NCONV_KF=2`
`PRLFLXCONV, PRSFLXCONV` : [3D] Liquid and Solid PRecipitation FLuX (m/s)	''

Turbulent scheme:	
TKEM : [3D] Turbulent Kinetic Energy (m^2/s^2)	LVAR_TURB
SIGS : [3D] σ_s from turbulence scheme (kg/kg^2)	‘ ’
SRCM : [3D] Normalized 2nd_order moment $\frac{\lambda_3 \overline{s'r'_c}}{2{\sigma_s}^2}$ (kg/kg^2)	‘ ’
BL_DEPTH : [3D] Boundary Layer Depth if CTOM='TM06' (m)	‘ ’
ATHETA, AMOIST : [3D] (m)	LTURBDIAG
RED_TH1, RED_R1, RED2_TH3, RED2_R3, RED2_THR3 : [3D]	
Redelsperger numbers (_)	‘ ’
DP, TP, TR, DISS: [3D] (m^2/s^3)	
dyn., therm. production, transport and dissipation of TKE	‘ ’
PHI3, PSI3, PSI_SV_n, LM, THLM, RNPM : [3D] (m,_,_,_,m,K,kg/kg)	‘ ’
• 1d-scheme turbulent fluxes:	
THW_FLX, RCONSW_FLX, RCW_FLX : [3D] (K m/s, kg/kg m/s)	LTURBFLX
THL_VVAR, THLRCONS_VCOR, RTOT_VVAR : [3D] (K^2, K kg/kg, $(kg/kg)^2$)	‘ ’
UW_VFLX, VW_VFLX : [3D] (m^2/s^2)	‘ ’
W_VVAR : [3D] (m^2/s^2)	‘ ’
WSV_FLX_n : [3D] (SVunit m/s)	‘ ’
• 3d-scheme turbulent fluxes:	
UTH_FLX, VTH_FLX, UR_FLX, VR_FLX : [3D] (K m/s, kg/kg m/s)	‘ ’
TH_HVAR, THR_HCOR, R_HVAR : [3D] (K^2,K kg/kg, $(kg/kg)^2$)	‘ ’
U_VAR, V_VAR, W_VAR : [3D] ($(m/s)^2$)	‘ ’
UV_FLX, UW_FLX, VW_FLX, USV_FLX_n : [3D] ($(m/s)^2$, SVunit m/s)	‘ ’
• Mixing length for clouds:	
RVCI : [3D] $rv + rc + ri$ (kg/kg)	LTURBDIAG
GX_RVCI,GY_RVCI : [3D] x and y gradient of RVCI (kg/kg/m)	‘ ’
GNORM_RVCI : [3D] Horizontal norm of the gradient of RVCI (kg/kg/m)	‘ ’
QX_RVCI,QY_RVCI : [3D] x and y gradient of	
the advection of RVCI (kg/kg/m)	‘ ’
QNORM_RVCI : [3D] Horizontal norm of	
the gradient of the advection of RVCI (kg/kg/m)	‘ ’
CEI : [3D] Cloud entrainment instability index (kg/kg/m/s)	‘ ’

Radiation scheme:	
DTHRAD : [3D] Radiative tendency for θ (K/s)	NRAD_3D=0,1,2,3,4,5
FLALWD, DIRFLASWD, SCAFLASWD, DIRSRFSWD: [2D] radiative fluxes (W/m^2)	''
CLEARCOL_TM1, EMIS: [2D] trace of cloud, emmissivity (_)	''
ZENITH,AZIM: [2D] solar zenithal angle, azimuthal angle (RAD)	''
DIR_ALB,SCA_ALB: [2D] direct albedo, scattered albedo (_)	''
TSRAD: [2D] radiative surface temperature (K)	''
SWF_DOWN, SWF_UP, LWF_DOWN: [3D] radiative fluxes (W/m^2)	NRAD_3D=1,2,3,4,5
LWF_UP, LWF_NET, SWF_NET: ''	''
DTRAD_LW, DTRAD_SW, : [3D] Radiative tendency for T (K/day)	''
RADSWD_VIS, RADSWD_NIR, RADLWD : [2D] surface radiative flux (W/m^2)	''
SWF_DOWN_CS, SWF_UP_CS, LWF_DOWN_CS: idem for Clear Sky results	NRAD_3D=2,3,4,5
LWF_UP_CS, LWF_NET_CS, SWF_NET_CS: ''	''
DTRAD_SW_CS, DTRAD_LW_CS : ''	''
RADSWD_VIS_CS, RADSWD_NIR_CS, RADLWD_CS : ''	''
PLAN_ALB_VIS, PLAN_ALB_NIR : [2D] planetary albedo (_)	NRAD_3D=3,4,5
PLAN_TRA_VIS, PLAN_TRA_NIR: [2D] planetary transmission (_)	''
PLAN_ABS_VIS, PLAN_ABS_NIR : [2D] planetary absorption (_)	''
EFNEB_DOWN, EFNEB_UP : [3D] equivalent emissivity (Morcrette scheme) (_)	NRAD_3D=4,5
FLWP, FIWP : [3D] liquid/ice water path (g/m^2)	''
EFRADL, EFRADI : [3D] cloud liquid water and ice effective radius (μm)	''
SW_NEB, RRTM_LW_NEB : [3D] (_) effective cloud fraction	''
OTH_VIS, OTH_NI1, OTH_NI2, OTH_NI3 : [3D] (_) optical thickness	''
SSA_VIS, SSA_NI1, SSA_NI2, SSA_NI3: [3D] (_) single scattering albedo	''
ASF_VIS, ASF_NIR1, ASF_NIR2, ASF_NIR3 : [3D] (_) asymetry factor	''
O3CLIM: [3D] (Pa/Pa) climatological ozone content	NRAD_3D=5
CUM_AER_LAND, CUM_AER_SEA ⎫ [3D] (_) cumulated optical thickness	''
CUM_AER_DES, CUM_AER_URB '' of the different aerosols	''
CUM_AER_VOL, CUM_AER_STRB ⎭'' from the top of the domain	''
Dust variables: (LDUST in YINIFILE.des)	
DSTM0nM: [3D] Dust 0-order moment of the lognormal mode n (ppbv)	
DSTM3nM: [3D] Dust 3rd-order moment of the lognormal mode n (ppbv)	
DSTM6nM: [3D] Dust 6rd-order moment of mode n (if LVARSIG) (ppbv)	
DSTRGAn: [3D] Dust number mean Radius of the lognormal mode n (μm)	
DSTRGAMn: [3D] Dust Mass mean Radius of the lognormal mode n (μm)	
DSTN0An: [3D] Dust Number of the lognormal mode n (/m^3)	
DSTSIGAn: [3D] Dust Standard deviation of the lognormal mode n (_)	
DSTMSSn: [3D] Dust Mass concentration of the lognormal mode n (μg/m^3)	
DSTBRDNn: [2D] Dust Burden of the lognormal mode n (g/m^2)	
DSTAOD2D: [2D] Dust Optical Depth (_)	NRAD_3D=1
DSTAOD3D: [3D] Dust Optical Depth between two vertical levels (_)	''
DSTEXT: [3D] Dust Extinction (1/km)	''
Chemical aerosol variables: (LORILAM in YINIFILE.des)	
SOAIM,... : [3D] Aerosol scalar variable (ppbv)	LCHEMDIAG
as defined in ch_aer_init_soa.f90	
RGAn: [3D] Aerosol number mean Radius of the lognormal mode n (μm)	''
RGAMn: [3D] Aerosol Mass mean Radius of the lognormal mode n (μm)	''
N0An: [3D] Aerosol Number of the lognormal mode n (/cc)	''
SIGAn: [3D] Aerosol Standard deviation of the lognormal mode n (_)	''

Variables [dim] meaning (unit)	in DIAG1.nam
Chemical variables: (LUSECHEM in YINIFILE.des)	
O3M,... : [3D] Chemical scalar variable (ppbv)	LCHEMDIAG
as defined in BASIC.f90	
TRAJectories using Lagrangian tracers: (LLG in YINIFILE.des)	
X, Y : [3D] X and Y coordinates (km)	LTRAJ
LGXM, LGYM, LGZM : [3D] Lagrangian tracers coordinates (m)	‘’
Xn, Yn, Zn : [3D] Lagrangian tracers coordinates at time origin n (km)	‘’
THn, RVn : [3D] corresponding Theta and Vapor mixing Ratio (K, g/kg)	‘’
i(see “Lagrangian analyses’ documentation” on Meso-NH web site (Books and Guides)	
RTTOV: Radiatif Transfer for Tiros Operational Vertical Sounder (RTTOV) code (version 8.7)	
PltSatSenBT: [2D] Brightness temperature (K)	NRTTOVINFO(1:4,nb)=
with Plt=Plateforme	PltId SatId SenId 0
Sat=Satellite	
Sen=Sensor	(1≤ nb≤ 10 BTs)
(see section6.3.2 for more details)	
Radiatif transfer code: (narrow-band code, (Morcrette, 1991))	
SAT_IRBT : [2D] Brightness temperature in IR channel (K)	CRAD_SAT=’SAT’
SAT_WVBT : [2D] and WV channel (K)	
for SAT in METEOSAT GOES-E INDSAT	
with subgrid condensation scheme taken into account (default .T.)	LRAD_SUBG_COND
(see section6.3.2 for more details)	
Using of RTTOV is highly recommended to compute brightness temperature.	

Variables [dim] meaning (unit)	in DIAG1.nam
MRV, MRC, MRR : [3D] Mixing Ratio for Vapor, Cloud, Rain (g/kg)	LVAR_MRW
MRI, MRS, MRG, MRH : ‘ ’ for Ice, Snow, Graupel, Hail (g/kg)	‘ ’
CCCNM, CCLOUDM, RAINM : [3D] C2R2 scalar variables (g/kg)	‘ ’
MRSVnnn : [3D] Mixing Ratio for User Scalar Variable n (g/kg)	LVAR_MRSV
TEMP, PRES : [3D] Temperature (C), Pressure (hPa)	LTPZH
ALT : [3D] height of model levels (geopotentiel in pressure level) (m)	‘ ’
REHU, VPRES : [3D] Relative Humidity (%), Vapor Pressure (hPa)	‘ ’
COREF, MCOREF : [3D] Refraction coindex, modified refraction coindex	LCOREF
THETAV : [3D] Virtual potential Temperature (K)	LMOIST_V
THETAE : [3D] Equivalent potential Temperature (K)	LMOIST_E
ABVOR : [3D] vertical component of Absolute Vorticity (/s)	LVORT
UM1, VM1, WM1 : [3D] relative vorticity components (/s)	‘ ’
POVOM : [3D] Potential Vorticity (PVU)	CISO=’EV’,’PREVTK’
POVOV : [3D] Virtual Potential Vorticity (PVU)	LMOIST_V
POVOE : [3D] Equivalent Potential Vorticity (PVU)	LMOIST_E
MEAN_POVO : [2D] Mean Potential Vorticity (PVU)	LMEAN_POVO
MEAN_POVOV : [2D] Mean Virtual Potential Vorticity (PVU)	‘ ’ , LMOIST_V
MEAN_POVOE : [2D] Mean Equivalent Potential Vorticity (PVU)	‘ ’ , LMOIST_E
averaged between two isobaric levels (in Pa)	XMEAN_POVO(1:2)
UM88, VM88, WM88 : [3D] Geostrophic wind components (m/s)	LGEO
UM89, VM89, WM89 : [3D] Ageostrophic wind components (m/s)	LAGEO
MSLP : [2D] Mean Sea Level Pressure (hPa)	LMSLP
THVW, THCW, THRW : [2D] Thickness of Vapor, Cloud and Rain Water (mm)	LTHW
THIC, THSN, THGR, THHA : [2D] ‘ ’ of ICe, SNow, GRaupel, HAil (mm)	‘ ’
HEC : [2D] Height of Explicit Cloud top (km)	LCLD_COV
HC, TC : [2D] maximum Cloud top: Height (km), Temperature (C)	‘ ’
CLDFR, VISI_HOR : [3D] Cloud Fraction (_), Visibility (m)	‘ ’
ACPRR, INPRR : [2D] ACcumulated/INstantaneous Precipitation Rates (explicit Rain) (mm,mm/h)	LVAR_PR
INPRR3D : [3D] INstantaneous 3D Rain Precipitation flux (explicit Rain) (m/s)	‘ ’
EVAP3D : [3D] INstantaneous 3D Rain Evaporation flux (explicit Rain) (kg/kg/s)	‘ ’
ACPRC, INPRC : [2D] ACcumulated/INstantaneous Precipitation Rates (explicit Cloud (mm,mm/h)	‘ ’
ACPRS, INPRS : [2D] ACcumulated/INstantaneous Precipitation Rates (explicit Snow) (mm,mm/h)	‘ ’
ACPRG, INPRG : [2D] ACcumulated/INstantaneous Precipitation Rates (Graupel) (mm,mm/h)	‘ ’
ACPRH, INPRH : [2D] ACcumulated/INstantaneous Precipitation Rates (Hail) (mm,mm/h)	‘ ’
ACPRT, INPRT : [2D] ACcumulated/INstantaneous Precipitation Rates (all explicit species) (mm,mm/h)	‘ ’
PACCONV, PRCONV : [2D] ACcumulated/instantaneous Precipitation Rates (total CONVective) (mm,mm/h)	‘ ’
PRSCONV : [2D] instantaneous Precipitation Rate (mm/h) (Solid CONVective)	‘ ’
ACTOPR : [2D] Accumulated Total Precipitation (mm)	LTOTAL_PR

Variables [dim] meaning (unit)	in DIAG1.nam
CAPEMAX : [2D] maximum of CAPE3D (J/kg)	NCAPE=0,1,2
CINMAX : [2D] value of CIN3D corresponding to CAPEMAX (J/kg)	‘’
CAPE3D, CIN3D, DCAPE3D : [3D] (J/kg)	NCAPE=1,2
VKE : [3D] Vertical Kinetic Energy (from explicit vertical motion) (J/kg)	NCAPE=2
BV, BVE : [3D] Brunt-Vaissala frequencies (/s)	LBV_FR
RARE : [3D] Radar Reflectivity (dBZ)	LRADAR
VDOP : [3D] radar Doppler fall speed (m/s)	‘’
ZDR : [3D] radar Differential Reflectivity (dBZ)	‘’
KDP : [3D] radar Differential Phase shift (degree/km)	‘’
HBLTOP: [2D] Height of boundary layer top (m)	LBLTOP
KBLTOP: [2D] Index of boundary layer top	‘’
FREE_ATM_GR: [2D] Gradient of free atmosphere above BL top (K/m)	‘’
THV_FREE: [3D] Thetav above BL top (K)	‘’
Surface parameters	
ZS : [2D] orography (m)	
ZSMT : [2D] smoothed orography for SLEVE vertical coordinate (m)	
Other surface variables are provided through the externalised surface namelists.	

6.3.1 Some formulae

Temperature (TEMP)

The temperature (in °C) is computed as:

$$T = \Pi.\theta - T_t \tag{6.1}$$

where T_t is the temperature of the triple point.

Vapor pressure and Relative humidity

The vapor pressure (VPRES) is computed as:

$$e = \frac{P}{1 + r_v.R_d/R_v} \tag{6.2}$$

The relative humidity (REHU) is computed as:

$$Hu = \frac{e}{e_s(T)} \tag{6.3}$$

When a mixed microphysical scheme is activated during the simulation, the saturation vapor pressure $e_s(T)$ is computed over ice at points where temperature is below the triple point.

Refraction coindexes (Hill, 1980)

The refraction coindex (`COREF`) is computed as:

$$N = (77.6/T).(P + 4810.e/T) - 6.e/T \tag{6.4}$$

where P and e are in hPa.
The modified refraction coindex (`MCOREF`) is computed as:

$$M = N + Z.10^6 a \tag{6.5}$$

where Z and a are respectively the altitude and the Earth radius in m.

Virtual potential temperature (`THETAV`)

$$\theta_v = \theta.\frac{(1 + r_v R_v/R_d)}{(1 + r_w)} \tag{6.6}$$

where r_w is the mixing ratio of total water substance

$$r_w = r_v + r_c + r_r + r_i + r_s + r_g + r_h$$

Equivalent potential temperature (`THETAE`)

The formulation is taken from Bolton (1980) following its equations (16), (21) and (43):

$$\begin{aligned} \theta_e &= \theta.exp\left[\Big(\frac{3376}{T_L} - 2.54\Big) r_v(1 + 0.81 r_v)\right] \\ \text{where } T_L &= \frac{2840}{3.5 \ln T - \ln e - 4.805} + 55 \\ \text{and } e &= \frac{0.01.P.r_v}{0.622 + r_v} \end{aligned} \tag{6.7}$$

Vorticity quantities

The relative vorticity (`UM1,VM1,WM1`) is computed as:

$$\begin{aligned} \vec{\zeta} = \vec{\nabla} \wedge \vec{U} &= (\frac{\partial w}{\partial \hat{y}} - \frac{\partial v}{\partial \hat{z}})\vec{i} \\ &\quad + (\frac{\partial u}{\partial \hat{z}} - \frac{\partial w}{\partial \hat{x}})\vec{j} \\ &\quad + (\frac{\partial v}{\partial \hat{x}} - \frac{\partial u}{\partial \hat{y}})\vec{k} \end{aligned} \tag{6.8}$$

The absolute vorticity (`ABVOR`) takes into account the rotation of the earth:

$$\xi = \vec{\zeta}.\vec{k} + 2\Omega \sin\varphi \tag{6.9}$$

Potential Vorticity (`POVOM`)

The Ertel potential vorticity is computed as:

$$P = \frac{\vec{\zeta}.\vec{\nabla}(\theta)}{\rho_{dref}} \tag{6.10}$$

The unit is the Potentiel Vorticity Unit, $1PVU = 10^6 K.m^2.kg^{-1}.s^{-1}$

Moist Potential Vorticities

The virtual potential vorticity (`POVOV`) is

$$P_v = \frac{\vec{\zeta}.\vec{\nabla}(\theta_v)}{\rho_{dref}} \tag{6.11}$$

and the equivalent virtual potential vorticity (`POVOE`)

$$P_e = \frac{\vec{\zeta}.\vec{\nabla}(\theta_e)}{\rho_{dref}} \tag{6.12}$$

Geostrophic and ageostrophic winds (`UM88,VM88, UM89,VM89`)

With the LHE system (see book1, chapter 2), the geostrophic wind is computed as:

$$u_g = -\frac{1}{f}.\frac{\partial(C_{pd}\theta_{vref}\Pi^{'})}{\partial\hat{y}} \quad ; \quad v_g = \frac{1}{f}.\frac{\partial(C_{pd}\theta_{vref}\Pi^{'})}{\partial\hat{x}} \tag{6.13}$$

With the MAE and DUR systems (see book1, chapter 2), the geostrophic wind is computed as:

$$u_g = -\frac{1}{f}.C_{pd}\theta_{vref}.\frac{\partial\Pi^{'}}{\partial\hat{y}} \quad ; \quad v_g = \frac{1}{f}.C_{pd}\theta_{vref}.\frac{\partial\Pi^{'}}{\partial\hat{x}} \tag{6.14}$$

where

$$\Pi^{'} = \left(\frac{P}{P_{00}}\right)^{\frac{R_d}{C_{pd}}} - \Pi_{ref}$$

The ageostrophic wind is computed as:

$$u_{ag} = u - u_g \quad ; \quad v_{ag} = v - v_g$$

Mean sea level pressure (`MSLP`)

The surface pressure is first computed as the mean between the pressure at the first mass level and at the level below. Then it is reduced to the mean sea level (where the height is zero) following the Laplace law:

$$P_{sea} = P_{surf}.exp\left(\frac{gz_{surf}}{R_d.T_v^m}\right) \tag{6.15}$$

where z_{surf} is the orography and T_v^m is the mean virtual temperature between the ground level and the sea level (the latter is extrapolated from the first with a climatological gradient of $-6.5K/km$).

Thickness of water species (THVW,THCW, THRW,THIC,THSN,THGR,THHA)

The thickness of a water specy x (with $x = v, c, r, i, s, g$ or h) is computed as:

$$\sum_{k=k_B}^{k=k_E} \frac{\rho_{dref}}{\rho_{liq.w.}}.r_x(k).\Delta z_k \tag{6.16}$$

Height of explicit cloud top (HEC)

For every columns scanned from the model top to the bottom, the height of explicit cloud top is the height where the cloud mixing ratio (r_c) exceeds the value of $0.1g/kg$. If a mixed microphysical scheme is activated during the simulation, the ice mixing ratio (r_i) is also taken into account (with the same threshold), and the height is the higher between the one of the top determined with r_c and the top determined with r_i.

Height and temperature of maximum cloud top (HC,TC)

If a convection scheme is activated during the simulation and if you ask for convective diagnostics (NCONV_KF≥0), the top of convective cloud computed by the convection scheme is compared to the previous one of explicit cloud in every columns. The height and the temperature of the higher top are deduced. For clear-sky columns, the height is 0 and the temperature is the one of the ground.

Visibility (VISI)

The visibility, function of the liquid water content, has not an universal formula. It is computed here for low level clouds according to Kunkel (1984) and the model COBEL :

$$VISI = \frac{3.9}{144.7(\frac{\rho_{dref} r_c}{1.+r_c})^{0.88}} \tag{6.17}$$

Height and index of boundary layer top (HBLTOP,KBLTOP)

The boundary layer top is found by checking $\partial\theta_v/\partial z$, and comparing it to the gradient between 5000m of height and the ground. It is the same algorithm than in PREP_REAL_CASE (used to shift variables when changing vertical grid), which has been found to work fairly well from polar to saharian area. More details can be found in the routine *free_atm_profile.f90* itself.

6.3.2 Comparison with satellite observations

A comparison between model outputs and satellite observations provides an assessment of how well the model can reproduce the meteorological situation. The model-to-satellite approach compares directly the satellite brightness temperatures (BTs) to the BTs computed from the predicted model fields (Morcrette, 1991). It has been first applied to Meso-NH outputs for

comparison with Meteosat observations in the infrared using a narrow-band code (Chaboureau et al., 2000). Since the Masdev4_7 version, the Radiative Transfer for Tiros Operational Vertical Sounder (RTTOV) code version 8.7 (Saunders et al., 2005) is also available allowing the calculation of BT for a large number of satellites.

Computing satellite images from a Meso-NH run

You need an additional file (300 Mo) which contains coefficients and the binary library of the transfer code itself, so:
- in the `DIAG1.nam` file, add the following lines

```
&NAM_DIAG
 NRTTOVinfo(:,1)= 3 6 20 0 /
FILESGET_LIST="rttov87_rtcoef.tar"
```

- in the `prepmodelrc` file, add the following line `LOAD_OPT="${MESONH}/binaries/librttov8.7.a"`

Here the first number (3) corresponds to the Meteosat platform, the second (6) to the satellite Meteosat-6, and the third (20) to the Seviri sensor (the fourth (0) is not used).

To simulate other instruments, use the code given in Tables 6.1 and 6.2 reproduced below from the RTTOV users guide (see `http://www.metoffice.com/research/interproj/nwpsaf/rtm/rtm_rttov8.html`).

Platform	RTTOVid	Satid range
NOAA	1	1 to 18
DMSP	2	8 to 16
Meteosat	3	5 to 7
GOES	4	8 to 12
GMS	5	5
FY-2	6	2 to 3
TRMM	7	1
ERS	8	1 to 2
EOS	9	1 to 2
METOP	*10*	*1 to 3*
ENVISAT	11	1
MSG	12	1 to 2
FY-1	13	3
ADEOS	14	1 to 2
MTSAT	15	1
CORIOLIS	16	1

Table 6.1: Platforms supported by RTTOV 8.7 as at 17 Nov 2005 in normal text. Platforms in italics are not yet supported by RTTOV 8.7 but soon will be. (taken from RTTOV 8.7 Users guide, Table 2, page 4).

Sensor	RTTOVid	Sensor Channel #	RTTOV-8 Channel #
HIRS	0	1 to 19	1 to 19
MSU	1	1 to 4	1 to 4
SSU	2	1 to 3	1 to 3
AMSU-A	3	1 to 15	1 to 15
AMSU-B	4	1 to 5	1 to 5
AVHRR	5	3b to 5	1 to 3
SSMI	6	1 to 7	1 to 4
VTPR1	7	1 to 8	1 to 8
VTPR2	8	1 to 8	1 to 8
TMI	9	1 to 9	1 to 9
SSMIS	10	1 to 24	1 to 21
AIRS	11	1 to 2378	1 to 2378
HSB	12	1 to 4	1 to 4
MODIS	13	1 to 17	1 to 17
ATSR	14	1 to 3	1 to 3
MHS	15	1 to 5	1 to 5
IASI	16	1 to 8461	1 to 8461
AMSR	17	1 to 14	1 to 7
MVIRI	20	1 to 2	1 to 2
SEVIRI	21	4 to 11	1 to 8
GOES-Imager	22	1 to 4	1 to 4
GOES-Sounder	23	1 to 18	1 to 18
GMS/MTSAT imager	24	1 to 4	1 to 4
FY2-VISSR	25	1 to 2	1 to 2
FY1-MVISR	26	1 to 3	1 to 3
CriS	27	TBD	TBD
CMISS	28	TBD	TBD
VIIRS	29	TBD	TBD
WINDSAT	30	1 to 10	1 to 5

Table 6.2: Instruments supported by RTTOV 8 7 as at 17 Nov 2005. Sensors in italics are not yet supported by RTTOV 8 7 but soon will be. (taken from RTTOV 8 7 Users guide, Table 3, page 5).

Handling satellite data into the Meso-NH world

Satellite data can be obtained from many different archive centers and with diffferent formats.

- One of them is the French archive center called SATMOS (Service d'Archivage et de Traitement Météorologique des Observations Spatiales; see `http://www.satmos.meteo.fr/`. SATMOS can provide observations from NOAA, GOES, GMS and METEOSAT satellites. See the quick look pages and send an email to satmos@meteo.fr to order satellite images (do not forget to specify the channels, the TARCYL format, the interest area, the date). The program `readtarcyl` below gives an example in reading the TARCYL format.

- An alternative way to get METEOSAT data is through EUMETSAT; see `http://www.eumetsat.int/`. Observation from NOAA satellites and many others can be also obtained CLASS (Comprehensive Large Array-data Stewardship System; see `http://www.class.noaa.gov/`). The ATOVS and AVHRR Pre-processing Package (AAPP; see `http://www.metoffice.com/research/interproj/nwpsaf/aapp/`) processes data from instruments on board the NOAA polar orbiting satellites - namely HIRS, AVHRR, AMSU and MHS on the current generation of satellites.

- Finally the satellite data can be projeted onto a Meso-NH grid by using the command `obs2mnh` from the libtools package.

Program readtarcyl:

```
PROGRAM readtarcyl
! ------------------------------------------------------
!    PURPOSE
!    -------
!     Read METEOSAT/MSG/GOES INFRARED data in TARCYL format
!     and write outputs to be readen by the obs2mesonh tool
!    AUTHOR
!    ------
!      J.-P. Chaboureau       *L.A.*
!    MODIFICATIONS
!    -------------
!      Original    08/03/06
! ------------------------------------------------------
USE F90_UNIX
IMPLICIT NONE
INTEGER :: II, IUNIT, IO, INARG, IUOUT, IUDIR
INTEGER :: JI, JJ, NBYTE, IXSIZE, IYSIZE, INIL, IVAL
REAL :: ZLON, ZLAT, ZVAL
REAL :: ZLATMIN, ZLATMAX, ZLONMIN, ZLONMAX, ZVALMIN, ZVALMAX
CHARACTER(LEN=80) :: YFNAM, YRAC, YLINE, YMEM, YDIR, YOUT
CHARACTER(LEN=21) :: YLBLCH
CHARACTER(LEN=12) :: YSAT
CHARACTER(LEN=2) :: YTMP2
CHARACTER(LEN=1) :: YTMP1

IUNIT=12
IUDIR=40
IUOUT=41
YMEM='def'
YDIR='satobs.dir'
YOUT='satobs.asc'
! ------------------------------------------------------
INARG = IARGC()
IF (INARG == 0) THEN
  PRINT *,'Usage: readtarcyl [tarcyl file]'
  STOP
END IF
CALL GETARG(1,YRAC)
! ------------------------------------------------------
YFNAM=TRIM(YRAC)//'.def'
```

```
OPEN(IUNIT,FILE=YFNAM,IOSTAT=IO)
IF (IO /= 0) THEN
  PRINT *,'PROBLEM WHILE OPENING '//YFNAM
  STOP
ELSE
  PRINT *,'OPEN '//TRIM(YFNAM)
ENDIF
YLBLCH='UNDX_XXX'
! -----------------------------------------------------
DO
  READ(IUNIT,'(A)',END=99) YLINE
  SELECT CASE ( YLINE(1:6) )
  CASE('SATIMG')
    YSAT=YLINE(8:20)
    IF (YSAT(1:8) == 'meteosat') YLBLCH(1:4)='MET'//YSAT(10:10)
    IF (YSAT(1:10) == 'meteosat08') YLBLCH(1:4)='MSG1'
    IF (YSAT(1:4) == 'goes') YLBLCH(1:4)='GO'//YSAT(5:6)
  CASE('NBYTE=')
    BACKSPACE(IUNIT)
    READ(IUNIT,26) YLINE(1:6),NBYTE
  CASE('XSIZE=')
    BACKSPACE(IUNIT)
    READ(IUNIT,26) YLINE(1:6),IXSIZE
  CASE('YSIZE=')
    BACKSPACE(IUNIT)
    READ(IUNIT,26) YLINE(1:6),IYSIZE
  CASE('LATMIN')
    YMEM=TRIM(YMEM)//' '//TRIM(YLINE)
    BACKSPACE(IUNIT)
    READ(IUNIT,27) YLINE(1:7),ZLATMIN
  CASE('LATMAX')
    YMEM=TRIM(YMEM)//' '//TRIM(YLINE)
    BACKSPACE(IUNIT)
    READ(IUNIT,27) YLINE(1:7),ZLATMAX
  CASE('LONMIN')
    YMEM=TRIM(YMEM)//' '//TRIM(YLINE)
    BACKSPACE(IUNIT)
    READ(IUNIT,27) YLINE(1:7),ZLONMIN
  CASE('LONMAX')
    YMEM=TRIM(YMEM)//' '//TRIM(YLINE)
    BACKSPACE(IUNIT)
    READ(IUNIT,27) YLINE(1:7),ZLONMAX
  END SELECT
  IF (YLINE(1:3) == 'ID=') THEN
    IF (YLBLCH(1:3)=='MET') THEN
      IF (YLINE(4:5) == 'ai') YLBLCH(6:8)='IR '
      IF (YLINE(4:5) == 'aw') YLBLCH(6:8)='WV '
    ELSEIF (YLBLCH(1:3)=='MSG') THEN
      YLBLCH(6:8)=YLINE(7:9)
    ELSEIF (YLBLCH(1:2)=='GO') THEN
      IF (YLINE(4:4) == '2') YLBLCH(6:8)='039'
      IF (YLINE(4:4) == '3') YLBLCH(6:8)='067'
      IF (YLINE(4:4) == '4') YLBLCH(6:8)='107'
      IF (YLINE(4:4) == '5') YLBLCH(6:8)='120'
    ENDIF
    PRINT *,TRIM(YLINE),' '//YLBLCH
  END IF
  IF (YLINE(1:3) == 'NIL') THEN
    BACKSPACE(IUNIT)
```

```
    READ(IUNIT,24) YLINE(1:4),INIL
  END IF
END DO
99 CLOSE(IUNIT)
24 FORMAT(A4,I8)
26 FORMAT(A6,I8)
27 FORMAT(A7,F4.0)
PRINT *,'NBYTE=',NBYTE,' IXSIZE=',IXSIZE,' IYSIZE=',IYSIZE
PRINT *, TRIM(YMEM)
! -----------------------------------------------------
YFNAM=TRIM(YRAC)//'.raw'
OPEN(IUNIT,FILE=YFNAM,FORM='UNFORMATTED', &
     access='direct',RECL=NBYTE,IOSTAT=IO)
IF (IO /= 0) THEN
  PRINT *,'PROBLEM WHILE OPENING '//YFNAM
  STOP
ELSE
  PRINT *,'OPEN '//TRIM(YFNAM)
ENDIF
! -----------------------------------------------------
II=0
ZVALMIN=999.
ZVALMAX=0.
OPEN(IUDIR,FILE=YDIR,FORM='FORMATTED')
! verbosity level
WRITE(IUDIR,'(I1)') 0
! format of obs file ll=lat lon alt value
WRITE(IUDIR,'(A2)') 'll'
! name of the obs file
WRITE(IUDIR,'(A12)') TRIM(YOUT)
! name of the obs field
WRITE(IUDIR,'(A21)') YLBLCH
! unit of the obs field
WRITE(IUDIR,'(A1)') 'K'
! profile of the obs field
WRITE(IUDIR,'(A2)') '2D'
! closing instruction
WRITE(IUDIR,'(A3)') 'END'
CLOSE(IUDIR)
OPEN(IUOUT,FILE=YOUT,FORM='FORMATTED')
IF (NBYTE==1) THEN
  DO JJ=1,IYSIZE
    DO JI=1,IXSIZE
      II=II+1
      READ(IUNIT,REC=II) YTMP1
      ZLON=ZLONMIN-JI*(ZLONMIN-ZLONMAX)/FLOAT(IXSIZE)
      ZLAT=ZLATMAX-JJ*(ZLATMAX-ZLATMIN)/FLOAT(IYSIZE)
      IVAL= ICHAR(YTMP1)
      IF (IVAL>0 .AND. IVAL<INIL) THEN
        ZVAL=FLOAT(IVAL)*150/255.+190
        ZVALMIN=MIN(ZVAL,ZVALMIN)
        ZVALMAX=MAX(ZVAL,ZVALMAX)
        WRITE(IUOUT,*) ZLAT,ZLON,0.,ZVAL
      END IF
    END DO
  END DO
ELSE
  DO JJ=1,IYSIZE
    DO JI=1,IXSIZE
```

```
      II=II+1
      READ(IUNIT,REC=II) YTMP2
      ZLON=ZLONMIN-JI*(ZLONMIN-ZLONMAX)/FLOAT(IXSIZE)
      ZLAT=ZLATMAX-JJ*(ZLATMAX-ZLATMIN)/FLOAT(IYSIZE)
      IVAL= ICHAR(YTMP2(1:1))*2**8 + ICHAR(YTMP2(2:2))
      IF (IVAL>0 .AND. IVAL<INIL) THEN
        ZVAL=FLOAT(IVAL)*0.01
        ZVALMIN=MIN(ZVAL,ZVALMIN)
        ZVALMAX=MAX(ZVAL,ZVALMAX)
        WRITE(IUOUT,*) ZLAT,ZLON,0.,ZVAL
      END IF
    END DO
  END DO
ENDIF
PRINT *,'VAL min/max ',ZVALMIN,ZVALMAX
CLOSE(IUNIT)
CLOSE(IUOUT)
! -----------------------------------------------------
END PROGRAM readtarcyl
```

6.4 References

Bolton, D., 1980: The Computation of Equivalent Potential Temperature. *Monthly Weather Review*, **108**, 1046-1053.

Chaboureau, J.-P., J.-P. Cammas, P. Mascart, J.-P. Pinty, C. Claud, R. Roca, and J.-J. Morcrette, 2000: Evaluation of a cloud life-cycle simulated by Meso-NH during FASTEX using METEOSAT radiances and TOVS-31 cloud retrievals. *Quart. J. Roy. Meteor. Soc.*, **126**, 1735-1750.

Hill R.J., 1980: Refractive index and absorption fluctuations in the infra red caused by temperature and pressure fluctuations. *J. Opt. Soc. Am.*, **70**, n$^c irc$10.

Kunkel, B.A., 1984 : Parameterization of droplet terminal velocity and extinction coefficient in fog models. *J. App. Meteor.* , **23**, 34-41.

Morcrette, J.-J., 1991: Radiation and cloud radiative properties in the European Center for Medium Weather Forecasts forecasting system. *J. Geophys. Res.*, **96**, 9121-9132.

Saunders, R., M. Matricardi, P. Brunel, S. English, P. Bauer, U. O'Keeffe, P. Francis and P. Rayer, 2005: RTTOV-8 science and validation report. NWP SAF Report, 41 pages, Tech. rep.

Chapter 7

Modify the Fortran sources :

7.1 Presentation of the different libraries

The source code of the model is managed by the UNIX standard source manager RCS (Revision Control System), present on Linux computers. With this source manager, you are able to get the source either from a reference library, i.e. the Master library or the Bugfix library, or from a personal library.

The Master and Bugfix sources build a safe set of Fortran 90 mains and subroutines, modified and updated by the administrator of the Meso-NH sources. We have seen in the previous chapters how to use the Meso-NH model as it stands, by collecting these binary objects, previously compiled on the "remote_host".

For your own purposes, you can add new sources or modify the Master sources by building your own library (the User's library). Eventually, some of these modifications may be incorporated in the next release of the Master library, after a merging with modifications from other users.

In this case, you will have to manipulate new objects with the procedures: the User's library on the "local_host" and its binary counterpart on the remote_host. You will have to compile these new sources (**prepsource** script) on the remote_host.

The first step is therefore to write the new Fortran 90 sources (Metcalf and Reed 1993) with an adaptation of the Doctor norm to the Fortran 90, described in Asencio et al.(1994). We refer to the Meso-NH web[1] site, in order to see how Meso-NH routines are written.

In the next section, we explain how to use the Meso-NH source file manager to store these files in a form, which allows its manipulation by the **prepsource** procedure (section 7.3). Then we show how to compile these sources (section 7.4) and how to use these binary objects in a simulation (section 7.5).

7.2 The Meso-NH source file management

The Meso-NH sources are managed using different elementary operations of the RCS source manager. Essentially, RCS maintains a source file by creating new versions with a different version number: 1.1 is the first version, then 1.2, 1.3 etc.... RCS also allows to extract, modify or delete old versions of the code. The specific tools developped for the Meso-NH system have basically the same functionalities of their RCS counterparts. But they also permit the

[1] `http://mesonh.aero.obs-mip.fr/mesonh`

identification of versions by character strings instead of the version identification number (VID), as in RCS.

This source file manager will help you to protect its developments against an unwanted destruction and to store different versions of the same source. It should be noted that you can create one library per Fortran source and collect all these libraries in the same directory called DIRSRC. This directory should also be different from the one SIMUL, from where the preparative scripts are run and which contains the namelists and rc files. This is not obligatory but strongly recommended to simplify the user's life. For instance, all the sources of the Master library are stored and protected by the same way but on more than one directory because of the large number of involved sources.

The Meso-NH source file manager updates the RCS file, by calling the appropriate RCS command. The basic RCS file contains the Fortran source code and all the necessary information necessary for RCS to recover the different versions. This file is often called the "file,v", since it is suffixed by ",v". The "file,v" also contains the necessary information to recover the RCS version corresponding to a Meso-NH version name.

Before using any Meso-NH source management tool, you may first initialize the environment variable DIRSRC, pointing to the absolute directory pathname, that contains either the Master, the Bugfix or the User library. The UNIX command is:

```
DIRSRC=/user/library ; export DIRSRC
```

If this environment variable is not initialized, its value is asked interactively by each Meso-NH RCS macro.

Table 7.1 lists all the Meso-NH macros for the source file management with their names, actions and RCS associated command. Of course, since most users are not allowed to modify the Master library sources, some macros cannot be used with the Master and Bugfix library but only with their private library (see last column).

macro	equiv. RCS command	action	library
`nh-create`	ci + tag	creates the file,v	UL
`nh-extract`	co	extracts a source in order to modify the library	UL
`nh-register`	ci + tag	modifies the source inside the library	UL
`nh-takeout`	co	extracts a source without intent to change the library	UL + ML
`nh-unextract`		undoes a previous nh-extract	UL
`nh-unregister`		undoes a previous nh-register	UL
`nh-story`		lists the historic for a given version name	UL + ML
`nh-target`		creates a mnh-file for a preexisting s-file	UL
`nh-show`		lists the mnh-files for a given version name	UL + ML

Table 7.1: The Meso-NH source file management tools: name, RCS associated command, brief description and library upon which it can be applied (UL for User's library and ML for Master and Bugfix libraries).

The full syntax of these macros is given in table 7.2.

full syntax	interpretation
`nh-create source.f90 VERS1`	creates source.f90,v where the version VERS1=1.1 is equal to source.f90
`nh-extract source.f90 VERS1`	extracts the last version (1.15, say) from the library to obtain a new version 1.16 and put it in the file source.f90
`nh-extract source.f90 VERS1 1.10`	extracts the version 1.10 from the library to obtain a new version 1.10.1.1 because 1.11 already exists
`nh-register source.f90 VERS1`	registers the modifications present in the file source.f90 in the new version of the library
`nh-takeout source.f90 VERS1`	takes out of the library, the VERS1 version of source.f90 and put it in the file source.f90
`nh-unextract source.f90 VERS1`	undoes the previous extraction; no new version is created
`nh-unregister source.f90 VERS1`	undoes the previous registration; the last version is removed
`nh-show VERS1`	shows the version identification numbers for all the files present in the library corresponding to the accronym VERS1
`nh-story source.f90 VERS1`	lists the historic of the source file starting from the RCS version that corresponds to the acronym VERS1
`nh-target source.f90 VERS2`	creates the acronym VERS2, the last version identification number is the one present in the file,v

Table 7.2: Full syntax of the Meso-NH macros and result of their actions.

Supplementary examples are given in the appendices of this book.

Global mode behaviour: the Meso-NH macros may also be used on a whole directory. In this case, the user must give a directory name instead of a file name in the macro arguments. Each macro will then perform its action on all source files in the directory. For example:
`nh-extract /users/come/fischer/master1 VERS1`
will extract all the source files associated to VERS1 and present in "master1". Furthermore, the source code is collected into a global source file named "master1.f90". Global mode behaviour is accepted by:

macro name	action
nh-extract/nh-takeout	extracts (takes out) source files and collects them into one file
nh-register	splits the collected source file and registers the modifications for each elementary file
nh-unextract	undoes a global nh-extract
nh-unregister	undoes a global nh-register
nh-story	lists the historic of all files

All these macros will help you take a look at the sources, to modify a source, register a modification either at the most basic level (file by file) or all the libraries present in the directory $DIRSRC at the same time. Of course, the Meso-NH user can start from a Fortran source taken from the Master library (`nh-takeout`), modify this Fortran source with any editor and create a RCS library enclosing its personal version of the Fortran source. He also can write from A to Z its Fortran source and enclose it in a RCS library. **It should be noted that all these actions are performed on the local_host**.

The next step is to compile on the remote_host these new or modified sources and to keep the binaries in binary library in order to avoid repeated compilations. This is achieved by the procedure named `prepsource`, presented in the next section.

7.3 The prepsource procedure

This procedure prepares on the local_host, a UNIX job, which contains all the orders necessary to achieve the compilation and storage of the results on the remote_host. All the Fortran sources have to be ready to be compiled, because the `prepsource` procedure does not allow you to modify these Fortran sources. These modifications had to be performed with the Meso-NH RCS procedures (`nh-create, nh-extract, nh-register`) in order to obtain RCS source libraries, which enclose the Fortran sources.

The main object of `prepsource` is to create or update a User's binary library on the remote_host. This library contains binary objects, which can be present or not in Master or Bugfix binary libraries (the generic term binary object applies either to a main program, a subroutine or a module). If a binary object is present in both libraries, the User's library one will be used when the model is loaded by `prepmodel`.

Because the different prognostic Meso-NH variables are defined in declarative modules (named MODD_...), the compiled form of this module must be available in the Master or User library when a Fortran source is compiled and uses these variables, transferred in this routine by a USE association (Metcalf and Reed 1993). The same problem occurs for the modules which enclose interfaces (named MODI_...). If the compiled form of the module is present in the User's library, it will be used during the compilations, else it will searched in the Master library. The last case occurs when the Meso-NH user compiles this module in the same job as other Fortran sources, which use it. In this case, `prepsource` first compiles all the MODI_..., MODD_... and in a second step, the other subroutines. Because the binaries of the User's library have priority on the Master's binaries, problems could occur, when you modify a MODD_.. present in the Master library without compiling **all** the sources which use this MODD_.. . The typical example is to add variable in a MODD_.., you must in this case, compile again **all** the sources which use this MODD_.., even if the new variable is not used!!. To prevent this type of problem, it is prohibited to modify a MODD_.. or a MODN_... which is present in the Master library (MODN_... is the

generic name for the module defining the Namelists). A control is realized by the `prepsource` procedure and if this rule is transgressed, the procedure stops and nothing is compiled.

Therefore, if you want to add a variable to a given module MODD_EXAMPLE, create a new module MODDB_EXAMPLE, which only contains the new variable. The proximity of the name MODDB_EXAMPLE with the pre-existing module, will help the Meso-NH administrator to merge the modifications coming from different Meso-NH users for the next release of the model.

Another way is to use the variables available in the module MODD_BLANK (also in MODN_BLANK present in all the MesoNH namelist files) by adding just the call to it (USE MODD_BLANK) in your subroutine, see the example of modeln.f90 in section 7.4.

To illustrate the locations of the libraries, we plot the different libraries and transfers on the figure 7.1.

The procedure `prepsource` is now described in detail, the control parameters are listed below :

- Parameters concerning the LOCAL_HOST:
  - MKFNAME: name for the model's makefile (default : $MESONH/procedures/make_mnh, practically always used)
- Parameters concerning the REMOTE_HOST:
  - BIBMASTER: file name for master binary library on the REMOTE_HOST
  - BIBBUGFIX: file name for bugfix binary library on REMOTE_HOST
  - BIBUSER: file name for the user's binary library on the REMOTE_HOST (starting at $HOME) or 0 if none
- Other control parameters:
  - DEBUG: compilation with debugging option or not (=debug or run) on the REMOTE_HOST
  - NSOURCE: number of copies for multi-tasked code (1 to 8)

Default values can be provided by a personal version of the prepsourcerc file which can be taken in the directory /mesonh/procedures.

There are eight possible configurations, depending on BIBMASTER, BIBBUGFIX and BIBUSER:

- BIBMASTER=something.a BIBBUGFIX=0 and BIBUSER=0. In this case, all object code that is not generated explicitly during the prepsource step will be extracted from the master binary library. This configuration allows to work only with the master library and no user code.
- BIBMASTER=something.a BIBBUGFIX=bugfix_name.a and BIBUSER=0. In this case, the object code will be extracted from the bugfix binary library if it is present, in the master binary library otherwise.

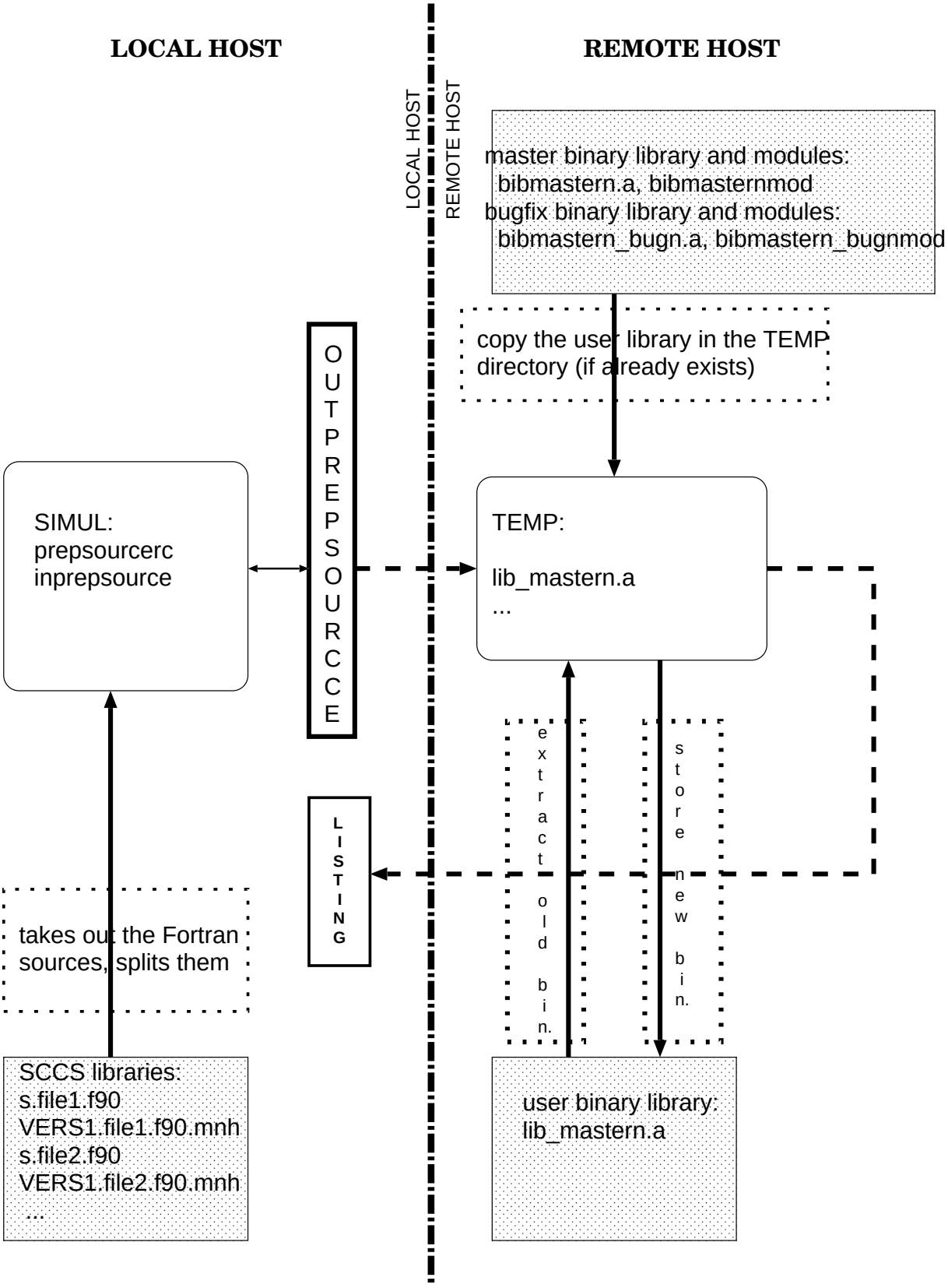


Figure 7.1: The different locations of the informations

- BIBMASTER=something.a BIBBUGFIX=0 and BIBUSER=anotherthing.a. This is the typical user case in which the new object code is stored in BIBUSER and missing object code will be extracted from BIBMASTER, without bugfix.

- BIBMASTER=something.a BIBBUGFIX=bugfix_name.a and BIBUSER=anotherthing.a. In this case, which should be a most common way of using the meso-nh procedures, the BIBUSER library is used (containing the modified source files objects) with bugfix.

- BIBMASTER=0 BIBBUGFIX=0 and BIBUSER=0. Here, the job will run with all the object code that is generated during the prepsource step. This code is then stored in a temporary library called templib.a that will not be saved. This is a precarious test configuration: all that will be left from the job is the output listing.

- BIBMASTER=0 BIBBUGFIX=0 and BIBUSER=something.a. This configuration allows to create a master library named something.a; this is useful only for the Master administrator.

- BIBMASTER=0 BIBBUGFIX=bugfix_name and BIBUSER=something.a. Incorrect case.

- BIBMASTER=0 BIBBUGFIX=bugfix_name and BIBUSER=0. Incorrect case.

`prepsource` performs the following actions:

1. it takes out the source files that are asked by the user (on LOCAL_HOST), for that, `prepsource` calls nh-takeout and the user must give to the procedure the name of the sccs libraries and the VID. An infinite loop is running, asking you if you want to "take out" a new file from a sccs library or to exit out of this loop. It should be noted that a whole set of sccs libraries, present in the directory with the same VID, can be processed all together by using the global mode of the procedure nh-takeout.

   If the source file, which contains the results of the nh-takeout procedure, already exists on your local directory, `prepsource` will ask you whether it can overwrite it or not.

2. the source is split into its elementary Fortran 90 modules (PROGRAM, SUBROUTINE, FUNCTION, MODULE).

3. Part of the model code is duplicated, when more than one model is used (grid-nesting active or NSOURCE=2 or more). In this case, the code is repeated NSOURCE times (maximum value for NSOURCE is 8), only changing the model number from one version to the other.

4. it also prepares the sequence of `make` commands in the right order for Fortran 90 (first modules, then sources) (actually performed on REMOTE_HOST) to compile the source files of step 1.

An example is presented in the next section, where a new subroutine is compiled with a modified source of the Meso-NH model.

7.4 An illustrative example of a source compilation

We first give the Fortran source of the new routine, in order to illustrate the Meso-NH "style" for the routine cfl, its interface encapsulated in a module (modi_cfl) and the subroutine model$n present in the Master library.

* FILE cfl.f90

```
!-----------------------------------------------------------------
!--------------- special set of characters for RCS information
!-----------------------------------------------------------------
! $Source: /home/training/stage/modif/cfl.f90,v $ $Revision: 1.1 $ $Date: 2002/03/21 08:56:04 $
!-----------------------------------------------------------------
!     ###############
      MODULE MODI_CFL
!     ###############
!
INTERFACE
      SUBROUTINE CFL ( PUT, PVT, PDXHAT, PDYHAT, PTSTEP, HLUOUT, TPDTCUR )
!
USE MODE_TIME
!
REAL, DIMENSION(:,:,:),   INTENT(IN)    :: PUT , PVT
                        ! 2 components of the horizontal Wind at t
REAL, DIMENSION(:),       INTENT(IN)    :: PDXHAT ! Stretching in x direction
REAL, DIMENSION(:),       INTENT(IN)    :: PDYHAT ! Stretching in y direction
REAL              ,       INTENT(IN)    :: PTSTEP ! time step
CHARACTER(LEN=*),         INTENT(IN)    :: HLUOUT !  Name of the output-listing
TYPE (DATE_TIME),         INTENT(OUT)   :: TPDTCUR! Current date and time
!
END SUBROUTINE CFL
END INTERFACE
!
END MODULE MODI_CFL
!
!     ######################################################################
      SUBROUTINE CFL ( PUT, PVT, PDXHAT, PDYHAT, PTSTEP, HLUOUT, TPDTCUR )
!     ######################################################################
!
!!****  *CFL * - compute the CFL number
!!
!!    PURPOSE
!!    -------
!!      The purpose of this routine is to compute the CFL number and to write a
!!    warning when its value is greater than 1.
!!
!!**  METHOD
!!    ------
!!
!!      The maximum value U of the horizontal wind is computed. Then the CFL
!!     number is computed according to:
!!          CFL= U DeltaT / DeltaX
!!       If this number is greater than 1, a warning message is written on the
!!     OUTPUT file together with the instant.
!!
!!    EXTERNAL
!!    --------
!!      Functions MXF,MYF   : computes the averages along the X and Y directions
!!      Subroutine FMLOOK   : to recover the Logical unit associated to a given File
```

```
!!
!!    IMPLICIT ARGUMENTS
!!    ------------------
!!      L2D in module MODD_CONF : switch for 2D configuration
!!      JPHEXT,JPVEXT in module MODD_PARAMETERS : number of external points
!!                    along the horizontal and the vertical directions
!!    REFERENCE
!!    ---------
!!      NONE
!!
!!    AUTHOR
!!    ------
!! J. Stein         * Meteo France *
!!
!!    MODIFICATIONS
!!    -------------
!!      Original    10/01/96
!------------------------------------------------------------------------------
!
!*       0.    DECLARATIONS
!              ------------
!
USE MODI_SHUMAN
!
USE MODE_TIME
USE MODE_FM
!
USE MODD_CONF
USE MODD_PARAMETERS
!
IMPLICIT NONE
!
!*       0.1   Declarations of dummy arguments :
!
REAL, DIMENSION(:,:,:),   INTENT(IN)    :: PUT , PVT
                        ! 2 components of the horizontal Wind at t
REAL, DIMENSION(:),       INTENT(IN)    :: PDXHAT ! Stretching in x direction
REAL, DIMENSION(:),       INTENT(IN)    :: PDYHAT ! Stretching in y direction
REAL              ,       INTENT(IN)    :: PTSTEP ! time step
CHARACTER(LEN=*),         INTENT(IN)    :: HLUOUT !  Name of the output-listing
TYPE (DATE_TIME),         INTENT(OUT)   :: TPDTCUR! Current date and time
!
!*       0.2   declarations of local variables
!
REAL    :: ZCFL      ! CFL number
REAL    :: ZWIND_MAX ! maximum value of the wind
REAL, DIMENSION(SIZE(PUT,1),SIZE(PUT,2),SIZE(PUT,3)) :: ZWIND_MOD ! modulus of
                       ! the horizontal wind
CHARACTER (LEN=40)     :: YTITLE                     !  Title for date print
INTEGER                :: ILUOUT, IRESP              !  Logical unit number
                        ! associated with HLUOUT and return code
INTEGER                :: IIB,IJB,IKB,IIE,IJE,IKE    ! index values for the
                                                     ! inner mass points
!------------------------------------------------------------------------------
!
!*       1.     COMPUTES THE MAXIMUM HORIZONTAL WIND
!         -----------------------------------
!
IIB=1+JPHEXT
```

```
IIE=SIZE(PUT,1) - JPHEXT
IJB=1+JPHEXT
IJE=SIZE(PUT,2) - JPHEXT
IKB=1+JPHEXT
IKE=SIZE(PUT,3) - JPVEXT
!
IF (.NOT. L2D) THEN
  ZWIND_MOD = SQRT( MXF(PUT)**2 + MYF(PVT)**2 )
  ZWIND_MAX = MAXVAL ( ZWIND_MOD(IIB:IIE,IJB:IJE,IKB:IKE) )
ELSE
  ZWIND_MAX = MAXVAL ( PUT(IIB:IIE,IJB:IJE,IKB:IKE) )
END IF
!
!-------------------------------------------------------------------------------
!
!*       2.     COMPUTES THE CFL NUMBER
!         ----------------------
!
IF (.NOT. L2D) THEN
  ZCFL = ZWIND_MAX * PTSTEP / MAX( MAXVAL(PDXHAT) , MAXVAL(PDYHAT) )
ELSE
  ZCFL = ZWIND_MAX * PTSTEP / MAXVAL(PDXHAT)
END IF
!
!-------------------------------------------------------------------------------
!
!*       3.     PRINTS THE WARNING FOR CFL>1
!         ---------------------------
!
IF ( ZCFL >= 1 ) THEN
  CALL FMLOOK_ll(HLUOUT,HLUOUT,ILUOUT,IRESP)
  YTITLE='WARNING FROM CFL'
  CALL SM_PRINT_TIME(TPDTCUR,HLUOUT,YTITLE)
  WRITE(ILUOUT,*) 'THE CFL NUMBER = ',ZCFL
  WRITE(ILUOUT,*) 'THE MAXIMUM HORIZONTAL WIND = ', ZWIND_MAX
END IF
!
!-------------------------------------------------------------------------------
!
END SUBROUTINE CFL
```

* FILE model_n.f90

```
!     ######spl
      SUBROUTINE MODEL_n(KTCOUNT, OEXIT)
!     ###################################
!
!!****  *MODEL_n * -monitor of the model version _n
....
!
USE MODD_BLANK
!
USE MODI_CFL
!
IMPLICIT NONE
...
!-------------------------------------------------------------------------------
!
!*       2.25bis CFL control
!
```

```
IF (.NOT.LDUMMY1) THEN
  CALL CFL(XUT, XVT, XDXHAT, XDYHAT, XTSTEP, CLUOUT, TDTCUR)
END IF
!
....
!-------------------------------------------------------------------------------
END SUBROUTINE MODEL_n
```

The first step is to create a directory, which will contain the RCS libraries and to go in this directory:

```
mkdir SOURCE
cd SOURCE
```

Create the new source files (cfl.f90 in our case) in this directory with any editor.

Modify the already-existing source files, first take it from the Master or Bugfix source libraries (for our case for $MESONH/sources/adiab or $MESONH/bugfix/mastern):

```
nh-takeout modeln.f90 MASDEV4_2
/mesonh/sources/adiab
```

(the "/mesonh/sources/adiab" is the answer to the question asked by the `nh-takeout` macro to determine the DIRSRC directory.)

Then, create your RCS library:

```
nh-create modeln.f90 EXAMPLE
.
nh-create cfl.f90 EXAMPLE
.
```

(the "." is the answer to the question asked by the `nh-create` macro to determine the DIRSRC directory.)

Then, the Meso-NH environment variables must be checked (refer to the previous example to prepare an initial file for an ideal case experiment). The `rc` files used for the compilation of these sources are: `prepsourcerc` and `tosupcrc`, or `mesonhrc`. We give below a copy of the resulting `rc` files.

FILE `prepsourcerc`

```
#! /bin/sh
#
# default initialization of environment variables for prepsource
#
# c. fischer 21/07/94
#-------------------------------------------------------------------------------
#
# instead of outprepsource
#OUTSCRIPT=
#
#      tori hpce tora lx32
#      a must have
OUTDEST=
#
# Automatic submit by tosupc, no tosupcrc required
LSOUMISAUTO=T
```

```
#
# name for master binary (file name)
# user remote binary libraries starting from $HOME
#
BIBMASTER=$DEFBIBMASTER
BIBBUGFIX=$DEFBIBBUGFIX
BIBUSER=modif.a
export BIBMASTER BIBBUGFIX BIBUSER
#
#####################################################################
#
# other variables: debugging option
# DEBUG = 'run' (everything else will be debug)
#
DEBUG=run
export DEBUG
#
# name of makefile if you've got a personal one
#
MKFNAME=make_mnh
export MKFNAME
#
# number of duplicate sources (spawning, multitasking)
#
NSOURCE=1
#
# global variables used in prepsource
#
#DEBUGSCRIPT=ON                          ;# ON /OFF
#ENVIRONMENT=                            ;# SILENTINTERACTIVE/INTERACTIVE/BATCH
```

FILE tosuprc

```
#! /bin/sh
#
# default input variables for tosupc (default values for the NQS parameters)
#
#  N. Asencio 13/12/94
#---------------------------------------------------------------------
#
# specific variables
#
                  # filename which contains the script to be run on remote-host
JOBFILE=${1:-0}
                  # jobname
JOBNAME=${2:-$(basename $JOBFILE)}

################### tosupc will not work unless you correctly fill TIME MEM TPN NBP ####
                  # time in seconds for J2&J3 jobs
TIME=
                  # memory ex: =2000 =2Gb =128Mb (tori: less than 128Gb per node)
                  # be careful : reduce the memory for multi-tasks jobs
MEM=
                  ### tasks per node: only for prep_ideal_case, run or diag
                  ### number of used CPUs from 1 to 8
  ### The most important for parallel execution!
TPN=1
                  # mono or multi-node on NEC (1 to 4) and on IBM
NBP=1
```

```
################### tosupc will not work unless you correctly fill TIME MEM TPN NBP ####

                  # Sending a mail abort + beg and/or at the end job (begend/beg/end/no)
                  # default is abort, =no is no mail at all
MAIL=

#
# global variables used in tosupc
#
#DEBUGSCRIPT=                              ; #ON /OFF
#ENVIRONMENT=SILENTINTERACTIVE             ; #SILENTINTERACTIVE/BATCH/INTERACTIVE
```

You are ready to run the procedure which prepares the compiling orders. Stay in the $SIMUL directory, where the `rc` files are. Work in ENVIRONMENT=INTERACTIVE (`export ENVIRONMENT=INTERACTIVE`) for the first time. Enter:

```
prepsource
```

The first question asked to you, is the localization of the SIMUL directory, answer "."

Then, a copy of the informations written in the `prepsourcerc` appears at the screen. You do not need to change anything. Validate the 2 pages with a blanck character after each page.

The second question concerns the localization of the RCS libraries (DIRSRC). The screen is as follows:

```
enter the root directory from which source files will be extracted
 default:
 1st :
 2nd : /users/mesonh/stein/SOURCE
 3rd : /mesonh/sources
 4th : /mesonh/bugfix/masdev4_1
?
```

The 2nd directory represents the desired directory DIRSRC and you only enter a carriage return. Then, the procedure asks you the name of the RCS library from where you want to take out the Fortran source.

```
please enter arguments for nh-takeout: file-name version-name [VID] ?
```

now, you enter one RCS library name:

```
cfl.f90 EXAMPLE
```

the answer refers to the RCS library cfl.f90,v and to the version defined by the EXAMPLE name. The procedure gives you the number of the desired version:

```
1.1
Retrieved:
1.1
436 lines
do you want another file: y or n ? (default y)
```

and asks you whether you want to compile another Fortran source or not. Enter :

```
y
```

You come back at the question where the directory for source location is prompted, answer by a carriage return and then by

```
modeln.f90 EXAMPLE
```

to take out the second Fortran source. Then, you go out of the infinite loop by answering n to the question. The `presource` procedure asks you :

```
enter the global parameters for compilation on remote machine  (makefile)
compilation options ? (default -c)
```

answer by a carriage return. Then `prepsource` works alone and splits the whole source in its Fortran elements and lists these elements:

```
==== end of (silent)interactive prepsource loop ====
==== a new file inprepsource.new is now available ==
==== for future work with prepsource in SILENTINTERACTIVE
modi_cfl.f90
cfl.f90
modeln1.f90
```

The `prepsource` step is now over and you will proceed as before with the `outprepsource` file. Send this file in the batch queues of the REMOTE_HOST, a report listing will be send back to the LOCAL_HOST, indicating the compilation report.

You can use a faster way of proceeding `prepsource` by using global mode for the `nh-takeout` procedure. The only difference appears when you answer the following question in `prepsource`:

```
please enter arguments for nh-takeout: file-name version-name [VID] ?
/users/come/stein/SOURCE EXAMPLE
```

Then, both files cfl.f90 and modeln.f90 will be taken out from the DIRSRC directory. This way of working is very interesting when you have a lot of sccs libraries present on the same directory and all managed with the same VID.

If compilation errors happen for one or more Fortran sources, you must correct the bugs in the sources files by creating new RCS versions (`nh-extract, nh-register`) on the LOCAL_HOST. Then, you can compile again these new sources using `prepsource`. Only the compilation of these sources is necessary, to complete your binary library on the REMOTE_HOST because the sources, where no bug were found, have been already saved in your binary library.

This time, you can work with `ENVIRONMENT=SILENTINTERACTIVE`. Just use the file `inprepsource.new` previously created by:

`cp inprepsource.new inprepsource`

modify it to keep only the source file(s) you want to compile again, **be careful of the syntax used in the file inprepsource** (each line corresponds to the informations put on the keyboard, blank included):

```
cfl.f90 EXAMPLE

modeln.f90 EXAMPLE
--
```

the first and third blanck lines correspond to the carriage return to the question of localisation of RCS sources (DIRSRC directory), the last one is for the carriage return to the additional compilation options, the -- characters are a separator.

Thus, working first in INTERACTIVE to produce an `inprepsource` file, and then modify it in SILENTINTERACTIVE, is an easy way to proceed with `prepsource`.

7.5 How to use a user's library

We have seen in the previous chapters how to use the Meso-NH model as it stands, by collecting the binary objects of the Master and the Bugfix source libraries, previously compiled on the "remote_host" by the Meso-NH administrator.

Once you have compiled your own source files (new subroutines created or pre-existing ones modified) and created on the remote host your User's binary library with their counterpart object files, you are ready to use them for the execution of one or several Meso-NH main programs. To do this, just add in the file `prepmodelrc`the name of your binary library present on the remote host in the control parameter `BIBUSER`. For the case of the previous example, file `prepmodelrc` looks like:

```
                              # user binary library
BIBUSER=modif.a
                              # reference binary library
BIBMASTER=masdev4_2.a
                              # reference bugfix binary library
BIBBUGFIX=bug1.a
```

Note that the name of the User's library created on the remote host is actually `modif_m4_2.a`. But it is not necessery to indicate the suffix (linked to the Master's binary name) added by `prepsource`, the `prepmodel` procedure will do the same operation to retrieve the right name of the User's library name.

Chapter 8

The MESONH files :

8.1 The F90 namelists

All the informations required to perform a given step of a numerical experiment, are provided by different files including a NAMELIST set. Thus, the Meso-NH user can change the value of the parameters without any compilation (and therefore save computer time). These files provide the way for the Meso-NH user to interact with the numerical code and finally, they contain the identification cards of the different steps of the numerical experiment.

These NAMELISTs are Fortran 90 NAMELISTs, which obey to strict writing rules (Metcalf and Reid 1993): no comment is allowed inside the namelists, no empty namelist can be written (it gives a Fortran execution error).

The informations are written in the following form :

```
&NAM_LUNITn  CINIFILE = "     FMFILE.1                " /
&NAM_CONFn  LUSERV = T, LUSERC = F, LUSERR = F, LUSERI = F, LUSERS = F,
LUSERG = F, LUSERH = F, NSV = 0 /
```

`&NAM_LUNITn` is the name of the first namelist of this file, the / character indicates the end of the list of informations. The parameters are valuated by VAR = VALUE and these prescriptions are separated one from each others by a comma and a blank character. Note that you can use more than one line to give one namelist, but in this case it is imperative to let a blank character at the end of each line.

The Meso-NH user does not need to prescribe all the parameters of one namelist, the missing informations are taken equal to the default values written in the fortran code. For example, the second namelist in the previous example can be written as:

```
&NAM_CONFn  LUSERV = T /
```

because the other variables of `&NAM_CONFn` are set to the default values.

In order to clearly separate what can be modified for a given step of a numerical experiment, we affect a different namelist file name for each step (see tab.2.1).

- To PREpare an Meso-NH file containing PhysioGraphical Datas $\Longrightarrow$ file PRE_PGD1.nam
- To PREpare an Meso-NH file with PhysioGraphical Datas in conformity $\Longrightarrow$ file PRE_NEST_PGD1.nam
- To PREpare an ZOOMed Meso-NH file with PhysioGraphical Datas $\Longrightarrow$ file PRE_ZOOM1.nam

- To PREpare an initial Meso-NH file for an IDEAlized case study $\Longrightarrow$ file PRE_IDEA1.nam
- To PREpare an initial Meso-NH file for an REAL case study $\Longrightarrow$ file PRE_REAL1.nam
- To SPAWN an Meso-NH file into another one with better horizontal resolution $\Longrightarrow$ file SPAWN1.nam
- To EXecute a simulation SEGment for the n^{th} model $\Longrightarrow$ file EXSEGn.nam
- To compute DIAGnostics after a simulation $\Longrightarrow$ file DIAG1.nam

Because the grid-nesting technic requires the simultanous presence of multiple models in the computer memory, free-parameters must be fixed for every model. This is performed by associating one namelist file per model, this explains why the namelist are suffixed by a number 1 or n just above.

The different parameters present in these files are all given in this book (chapters 3 to 6) and more details on the description of a given parameter are present in the code itself.

8.2 The Meso-NH files

A Meso-NH FM-file is a set of 2 files, sticked together via the cpio UNIX command, the 2 parts are:

- a descriptive part (`.des`) containing informations about the file generation and its description (in ASCII characters)
- a binary part (`.lfi`) where the fields are stored. The structure of this file is a direct access type file, written and read by routines developped in Météo-France (Fischer, 1994) based on LFI routines (Clochard, 1989), which can be used on a lot of different computers.

This type of file is used to store all the data necessary to run any step of a numerical experiment. Three different files are taken into account in the Meso-NH project:

- the synchronous file contains all the values of all the fields allowing a restart of the model and of some diagnostic fields desired by the Meso-NH user. All these informations are obtained at the same instant during the simulation, thus they are synchronous.
- the diachronic file contains time series of informations desired by the Meso-NH user. They are obtained during more than one time step of the model. It is the format in which your file must be in you want to plot it with the graphics software `diaprog` (you can convert a synchronous file into a diachronic one with `conv2dia`).
- the physiographic file contains external informations like orography, vegetation classes, chemical emissions, data sets, etc.

8.2.1 The synchronous file

This type of file contains only informations corresponding to the same instant of the simulation, it remains open during a whole time step of the simulation, and the writing orders can be given from any routine of the model.

The descriptive part

This part is the list of all the namelists of the EXSEG$n.nam file. Thus, a complete description of this part is given with the EXSEG$n.nam description in chapter 4.

If the file has been generated during a segment of the model integration, the `.des` part contains the different namelists fixing the free-parameters for the dynamics and the physics of the Meso-NH model. This allows the user to know a large part of the history of this file. For the namelists or variables ommited in the EXSEG$n.nam file, the values are set to the default ones (see the tables in ch.4).

If the file is the result of the initialization programs (`PREP_IDEAL_CASE`, `PREP_REAL_CASE` or `SPAWNING`), the values of the namelists variables are the ones of the descriptive part of the input file of the program if it does exist. Otherwise, the values are set to the default ones, except for these that can be initialized during the initialization program (e.g. `CINIFILE` or `LUSERV`).

Note that a physiographic file does not have a descriptive part.

The binary part

All the writings and readings of this type of files are done through LFI routines. A general subroutine to read and write a Meso-NH file is given in the Meso-NH library, it provides a file including the fields of the previous record list. This Fortran library provides a way to tackle direct access binary files and thus a very quick access to the data stored in this file in any order.

It should be noted that supplementary fields can be added to these basic informations, which have been obtained at the same instant. In order to be easily drawn by the Meso-NH graphic package, the commentar field must be filled, according to the following rules:

- the length of the character string is equal to 100
- the type of the supplementary field must be specified :

type	commentar field
3D scalar	X_Y_Z_varname (UNIT)
2D scalar	X_Y_varname (UNIT)
3D vector	VX_xvarname_VY_yvarname_VZ_zvarname (UNIT)
2D scalar	VX_xvarname_VY_yvarname_VZ_zvarname (UNIT) or VX_xvarname_VY_yvarname (UNIT)
1D scalar	Z_zvarname (UNIT)

Content of a synchronous file with *fmmore*

As explained in section 5.10.1, this procedure gives information about the MesoNH file. First, it lists the records present in the binary part, then the values of some parameters present in the binary part (e.g. the dimensions of the fields), and finally the descriptive part of the file. This set of informations can be obtained for any Meso-NH file, we give here an example for a synchronous file. You may use the procedure `fmmore` by the following order:

```
fmmore FILENAME
```

```
 81,  1,  256,  125696

 ///// LFILAF - Catalogue de l'Unite Logique LFI 12 dans l'ordre *PHYSIQUE* (sequentiel) des articles

  3 article(s) "physique(s)" de gestion,   512 mots chacun, occupant donc   1536 mots; detail:
      Article documentaire de la position 1 a   512
```

```
        1 paire(s) d'articles d'index prereserves, de la position   513 a   1536
    -----   pas de paire d'articles d'index inutilises ni excedentaires   -----


 1-eme article de donnees: "MY_NAME        ",     50 mots, position     1537 a     1586
 2-eme article de donnees: "DAD_NAME       ",     50 mots, position     1587 a     1636
 3-eme article de donnees: "STORAGE_TYPE   ",     24 mots, position     1637 a     1660
 4-eme article de donnees: "IMAX           ",     23 mots, position     1661 a     1683
 5-eme article de donnees: "JMAX           ",     23 mots, position     1684 a     1706
 6-eme article de donnees: "KMAX           ",     23 mots, position     1707 a     1729
 7-eme article de donnees: "THINSHELL      ",     23 mots, position     1730 a     1752
 8-eme article de donnees: "LAT0           ",     23 mots, position     1753 a     1775
 9-eme article de donnees: "LON0           ",     23 mots, position     1776 a     1798
10-eme article de donnees: "BETA           ",     23 mots, position     1799 a     1821
11-eme article de donnees: "XHAT           ",    184 mots, position     1822 a     2005
12-eme article de donnees: "YHAT           ",     25 mots, position     2006 a     2030
13-eme article de donnees: "ZHAT           ",     68 mots, position     2031 a     2098
14-eme article de donnees: "ZS             ",    508 mots, position     2099 a     2606
15-eme article de donnees: "DTCUR%TDATE    ",     25 mots, position     2607 a     2631
16-eme article de donnees: "DTCUR%TIME     ",     23 mots, position     2632 a     2654
17-eme article de donnees: "DTEXP%TDATE    ",     25 mots, position     2655 a     2679
18-eme article de donnees: "DTEXP%TIME     ",     23 mots, position     2680 a     2702
19-eme article de donnees: "DTMOD%TDATE    ",     25 mots, position     2703 a     2727
20-eme article de donnees: "DTMOD%TIME     ",     23 mots, position     2728 a     2750
21-eme article de donnees: "DTSEG%TDATE    ",     25 mots, position     2751 a     2775
22-eme article de donnees: "DTSEG%TIME     ",     23 mots, position     2776 a     2798
23-eme article de donnees: "CARTESIAN      ",     23 mots, position     2799 a     2821
24-eme article de donnees: "UM             ",  22378 mots, position     2822 a    25199
25-eme article de donnees: "VM             ",  22378 mots, position    25200 a    47577
26-eme article de donnees: "WM             ",  22378 mots, position    47578 a    69955
27-eme article de donnees: "THM            ",  22378 mots, position    69956 a    92333
28-eme article de donnees: "TKEM           ",  22378 mots, position    92334 a   114711
29-eme article de donnees: "PABSM          ",  22378 mots, position   114712 a   137089
30-eme article de donnees: "RVM            ",  22378 mots, position   137090 a   159467
31-eme article de donnees: "RCM            ",  22378 mots, position   159468 a   181845
32-eme article de donnees: "RRM            ",  22378 mots, position   181846 a   204223
33-eme article de donnees: "RIM            ",  22378 mots, position   204224 a   226601
34-eme article de donnees: "RSM            ",  22378 mots, position   226602 a   248979
35-eme article de donnees: "RGM            ",  22378 mots, position   248980 a   271357
36-eme article de donnees: "LSUM           ",  22378 mots, position   271358 a   293735
37-eme article de donnees: "LSVM           ",  22378 mots, position   293736 a   316113
38-eme article de donnees: "LSWM           ",  22378 mots, position   316114 a   338491
39-eme article de donnees: "LSTHM          ",  22378 mots, position   338492 a   360869
40-eme article de donnees: "LSRVM          ",  22378 mots, position   360870 a   383247
41-eme article de donnees: "LSXTKEM        ",    298 mots, position   383248 a   383545
42-eme article de donnees: "LSYTKEM        ",  14926 mots, position   383546 a   398471
43-eme article de donnees: "LSXRCM         ",    298 mots, position   398472 a   398769
44-eme article de donnees: "LSYRCM         ",  14926 mots, position   398770 a   413695
45-eme article de donnees: "LSXRRM         ",    298 mots, position   413696 a   413993
46-eme article de donnees: "LSYRRM         ",  14926 mots, position   413994 a   428919
47-eme article de donnees: "LSXRIM         ",    298 mots, position   428920 a   429217
48-eme article de donnees: "LSYRIM         ",  14926 mots, position   429218 a   444143
49-eme article de donnees: "LSXRSM         ",    298 mots, position   444144 a   444441
50-eme article de donnees: "LSYRSM         ",  14926 mots, position   444442 a   459367
51-eme article de donnees: "LSXRGM         ",    298 mots, position   459368 a   459665
52-eme article de donnees: "LSYRGM         ",  14926 mots, position   459666 a   474591
53-eme article de donnees: "UT             ",  22378 mots, position   474592 a   496969
54-eme article de donnees: "VT             ",  22378 mots, position   496970 a   519347
55-eme article de donnees: "WT             ",  22378 mots, position   519348 a   541725
```

```
56-eme article de donnees: "THT            ",  22378 mots, position   541726 a   564103
57-eme article de donnees: "TKET           ",  22378 mots, position   564104 a   586481
58-eme article de donnees: "RVT            ",  22378 mots, position   608860 a   631237
59-eme article de donnees: "RCT            ",  22378 mots, position   631238 a   653615
60-eme article de donnees: "RRT            ",  22378 mots, position   653616 a   675993
61-eme article de donnees: "RIT            ",  22378 mots, position   675994 a   698371
62-eme article de donnees: "CIT            ",  22378 mots, position   698372 a   720749
63-eme article de donnees: "RST            ",  22378 mots, position   720750 a   743127
64-eme article de donnees: "RGT            ",  22378 mots, position   743128 a   765505
65-eme article de donnees: "DRYMASST       ",     23 mots, position   765506 a   765528
66-eme article de donnees: "SRCM           ",  22378 mots, position   765529 a   787906
67-eme article de donnees: "SRCT           ",  22378 mots, position   787907 a   810284
68-eme article de donnees: "SIGS           ",  22378 mots, position   810285 a   832662
69-eme article de donnees: "RHOREFZ        ",     68 mots, position   832663 a   832730
70-eme article de donnees: "THVREFZ        ",     68 mots, position   832731 a   832798
71-eme article de donnees: "EXNTOP         ",     23 mots, position   832799 a   832821
72-eme article de donnees: "INPRR          ",    508 mots, position   832822 a   833329
73-eme article de donnees: "ACPRR          ",    508 mots, position   833330 a   833837
74-eme article de donnees: "INPRS          ",    508 mots, position   833838 a   834345
75-eme article de donnees: "ACPRS          ",    508 mots, position   834346 a   834853
76-eme article de donnees: "INPRG          ",    508 mots, position   834854 a   835361
77-eme article de donnees: "ACPRG          ",    508 mots, position   835362 a   835869
78-eme article de donnees: "INPRT          ",    508 mots, position   835870 a   836377
79-eme article de donnees: "ACPRT          ",    508 mots, position   836378 a   836885
80-eme article de donnees: "PABST          ",  22458 mots, position   836886 a   859343
81-eme article de donnees: "NEB            ",  22458 mots, position   859344 a   881801 <    +375 >

    --------        81 articles logiques de donnees et     1 trous repertories listes   --------

 ///// LFILAF - Fin du catalogue de l'Unite Logique 12 ---     82 Articles logiques en tout


 ###################################################################
 ################        COMMENTS      ##########################
 ###################################################################
 ###################################################################
 ####  CSTORAGE_TYPE = MT
 ####
 ####  CMY_NAME = COPTE.1.NEWIC.007
 ####
 ####  CDAD_NAME=
 ####
 ####  NIMAX =  160      NJMAX =  1      NKMAX =  44
 ####
 ####  LCARTESIAN =  T
 ####
 ####  LTHINSHELL =  T
 ####
 ####  XBETA =  0.E+0
 ####
 ####  XLAT0 =  43.28999999999996
 ####
 ####  x mesh =  2000.
 ####
 ####  y mesh =  2000.
 ####
 ####  XZHAT(  1) =   -220.49000
 ####  XZHAT(  2) =      0.00000
 ####  XZHAT(  3) =    220.49000
```

```
 ####  XZHAT(  4) =    444.46000
 ####  XZHAT(  5) =    672.53000
 ####  XZHAT(  6) =    904.91000
 ####  XZHAT(  7) =   1141.80000
 ####  XZHAT(  8) =   1383.40000
 ####  XZHAT(  9) =   1629.90000
 ####  XZHAT( 10) =   1881.70000
 ####  XZHAT( 11) =   2138.90000
 ####  XZHAT( 12) =   2401.90000
 ####  XZHAT( 13) =   2670.90000
 ####  XZHAT( 14) =   2946.40000
 ####  XZHAT( 15) =   3228.60000
 ####  XZHAT( 16) =   3518.10000
 ####  XZHAT( 17) =   3815.20000
 ####  XZHAT( 18) =   4120.50000
 ####  XZHAT( 19) =   4434.50000
 ####  XZHAT( 20) =   4757.80000
 ####  XZHAT( 21) =   5091.00000
 ####  XZHAT( 22) =   5435.10000
 ####  XZHAT( 23) =   5790.80000
 ####  XZHAT( 24) =   6158.90000
 ####  XZHAT( 25) =   6540.70000
 ####  XZHAT( 26) =   6937.30000
 ####  XZHAT( 27) =   7350.10000
 ####  XZHAT( 28) =   7780.80000
 ####  XZHAT( 29) =   8231.20000
 ####  XZHAT( 30) =   8703.30000
 ####  XZHAT( 31) =   9199.80000
 ####  XZHAT( 32) =   9723.80000
 ####  XZHAT( 33) =  10278.00000
 ####  XZHAT( 34) =  10869.00000
 ####  XZHAT( 35) =  11499.00000
 ####  XZHAT( 36) =  12175.00000
 ####  XZHAT( 37) =  12905.00000
 ####  XZHAT( 38) =  13697.00000
 ####  XZHAT( 39) =  14563.00000
 ####  XZHAT( 40) =  15512.00000
 ####  XZHAT( 41) =  16557.00000
 ####  XZHAT( 42) =  17706.00000
 ####  XZHAT( 43) =  18954.00000
 ####  XZHAT( 44) =  20282.00000
 ####  XZHAT( 45) =  21646.00000
 ####  XZHAT( 46) =  22515.00000
 ####
 ####  TDTCUR%TDATE =  1993,  2,  23
 ####  TDTCUR%TIME =  18000.
 ####
 ####  TDTEXP%TDATE =  1993,  2,  22
 ####  TDTEXP%TIME =  79200.
 ####
 ####  TDTMOD%TDATE =  1993,  2,  22
 ####  TDTMOD%TIME =  79200.
 ####
 ####  TDTSEG%TDATE =  1993,  2,  22
 ####  TDTSEG%TIME =  79200.
 ####
 KTYPE= 0
 YCOMMAND=transfer.x NIL            COPTE.1.NEWIC.007
-------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------
CONTENT  OF  THE  DESCRIPTIVE  PART  OF  FILE  COPTE.1.NEWIC.007
-------------------------------------------------------------------------
 &NAM_LUNITN  CINIFILE = "COPT                          " ,
  CCPLFILE(1) = "                               " ,
  CCPLFILE(2) = "                               " ,
  CCPLFILE(3) = "                               " ,
  CCPLFILE(4) = "                               " ,
  CCPLFILE(5) = "                               " ,
  CCPLFILE(6) = "                               " ,
  CCPLFILE(7) = "                               " ,
  CCPLFILE(8) = "                               " /
 &NAM_CONFN  LUSERV = T, LUSERC = T, LUSERR = T, LUSERI = T, LUSERS = T, LUSERG = T,
LUSERH = F, LUSECI = T, NSV = 0 /
 &NAM_DYNN  XTSTEP =    7.500 , CPRESOPT = "RICHA" , NITR =   4 ,
   XRELAX = 1.0000 , LHO_RELAX = F , LVE_RELAX = T , NRIMX =   4 ,
   NRIMY =   4 , XRIMKMAX = 0.00050000 , XT4DIFF =  1000.0000 /
 &NAM_ADVN  CMET_ADV_SCHEME = "FCT2ND", CSV_ADV_SCHEME = "FCT2ND",
CFV_ADV_SCHEME = "MPDATA", NLITER = 2 /
 &NAM_PARAMN  CTURB = "TKEL", CRAD = "NONE", CDRAG = "NONE", CGROUND = "FLUX",
 CCLOUD = "ICE3", CDCONV = "NONE" /
 &NAM_PARAM_RADN  XDTRAD = 60., XDTRAD_CLONLY = 60., LCLEAR_SKY = F,
NSPOT = 1, NRAD_COLNBR = 1000 /
 &NAM_PARAM_CONVECTN  XDTCONV = 300., NICE = 1 /
 &NAM_PARAM_GROUNDN  CROUGH = "Z04D" /
 &NAM_LBCN  CLBCX = 2*"OPEN", CLBCY = 2*"CYCL", NLBLX = 2*1, NLBLY = 2*1,
XCPHASE = 20. /
 &NAM_TURBN  XIMPL = 0.E+0, CTURBLEN = "DELT", CTURBDIM = "3DIM",
LTURB_FLX = F, LTURB_DIAG = F, LSUBG_COND = F /
 &NAM_CH_MNHCN  LUSECHEM = F, LCH_INIT_FIELD = F, LCH_SURFACE_FLUX = F,
 CCH_INIT_FIELD_OPT = "NONE
 ", CCH_SURFACE_FLUX_OPT = "NONE                                    ", CCHEM_INPUT_FILE = "MNHC.input
                                  ", CCH_TDISCRETIZATION = "SPLIT     ", NCH_SUBSTEPS = 1 /
 &NAM_CONF  CCONF = "START", L2D = T, L1D = F, LFLAT = T, NMODEL = 1, CEQNSYS = "DUR",
 NVERB = 1, CEXP = "COPTE", CSEG = "NEWIC", LFORCING = F /
 &NAM_DYN  XSEGLEN = 28807.5, XASSELIN = 0.2000000000000002, LCORIO = F,
LNUMDIFF = T, LSTEADYLS = T, XALKTOP = 5.000000000000004E-3, XALZBOT = 17000. /
 &NAM_NESTING  NDAD = 1,  2,  3,  4,  5,  6,  7,  8, NDTRATIO = 8*1, XWAY = 0.E+0,  7*2. /
 &NAM_FMOUT  XFMOUT = 8*3600.,  8*7200.,  8*10800.,  8*14400.,  8*18000.,  8*21600.,  8*25200.,  8*28800.,  96*-
 &NAM_BUDGET  CBUTYPE = "NONE", NBUMOD = 1, XBULEN = 300., NBUKL = 2, NBUKH = 46, LBU_KCP = F, XBUWRI = 300., NB
  NBUJL = 3, NBUJH = 18, LBU_ICP = T, LBU_JCP = T, NBUMASK = 2 /
 &NAM_BU_RU  LBU_RU = F, NASSEU = 1, NADVXU = 1, NADVYU = 1, NADVZU = 1, NFRCU = 0, NCURVU = 1, NCORU = 1, NDIFU
  NHTURBU = 1, NVTURBU = 1, NPRESU = 1 /
 &NAM_BU_RV  LBU_RV = F, NASSEV = 1, NADVXV = 0, NADVYV = 0, NADVZV = 0, NFRCV = 0, NCURVV = 0, NCORV = 0, NDIFV
  NHTURBV = 0, NVTURBV = 0, NPRESV = 0 /
 &NAM_BU_RW  LBU_RW = F, NASSEW = 1, NADVXW = 0, NADVYW = 0, NADVZW = 0, NFRCW = 0, NCURVW = 0, NCORW = 0, NGRAV
  NRELW = 0, NHTURBW = 0, NVTURBW = 0, NPRESW = 0 /
 &NAM_BU_RTH  LBU_RTH = F, NASSETH = 1, NADVXTH = 1, NADVYTH = 1, NADVZTH = 1, NFRCTH = 0, NPREFTH = 1, NDIFTH =
  NRADTH = 0, NDCONVTH = 0, NHTURBTH = 1, NVTURBTH = 1, NREVATH = 1, NCONDTH = 1, NHENUTH = 0, NDEPSTH = 0, NDEP
  NACCTH = 0, NCFRZTH = 0, NWETGTH = 0, NDRYGTH = 0, NGMLTTH = 0, NIMLTTH = 0, NBERFITH = 0, NCDEPITH = 0 /
 &NAM_BU_RTKE  LBU_RTKE = F, NASSETKE = 1, NADVXTKE = 0, NADVYTKE = 0, NADVZTKE = 0, NFRCTKE = 0, NDIFTKE = 0, N
  NDISSTKE = 0, NTRTKE = 0 /
 &NAM_BU_RRV  LBU_RRV = F, NASSERV = 1, NADVXRV = 0, NADVYRV = 0, NADVZRV = 0, NFRCRV = 0, NDIFRV = 0, NRELRV =
  NHTURBRV = 0, NVTURBRV = 0, NREVARV = 0, NCONDRV = 0, NHENURV = 0, NDEPSRV = 0, NDEPGRV = 0, NCDEPIRV = 0 /
 &NAM_BU_RRC  LBU_RRC = F, NASSERC = 1, NADVXRC = 0, NADVYRC = 0, NADVZRC = 0, NFRCRC = 0, NDIFRC = 0, NDCONVRC
  NVTURBRC = 0, NACCRRC = 0, NAUTORC = 0, NCONDRC = 0, NRIMRC = 0, NDRYGRC = 0, NIMLTRC = 0, NBERFIRC = 0, NCDEP
 &NAM_BU_RRR  LBU_RRR = F, NASSERR = 1, NADVXRR = 0, NADVYRR = 0, NADVZRR = 0, NFRCRR = 0, NDIFRR = 0, NACCRRR =
  NREVARR = 0, NSEDIRR = 0, NACCRR = 0, NCFRZRR = 0, NWETGRR = 0, NDRYGRR = 0, NGMLTRR = 0 /
 &NAM_BU_RRI  LBU_RRI = F, NASSERI = 1, NADVXRI = 0, NADVYRI = 0, NADVZRI = 0, NFRCRI = 0, NDIFRI = 0, NDCONVRI
```

```
 NHENURI = 0, NAGGSRI = 0, NAUTSRI = 0, NCFRZRI = 0, NWETGRI = 0, NDRYGRI = 0, NIMLTRI = 0, NBERFIRI = 0, NCDEPIRI =
&NAM_BU_RRS  LBU_RRS = F, NASSERS = 1, NADVXRS = 0, NADVYRS = 0, NADVZRS = 0, NFRCRS = 0, NDIFRS = 0, NSEDIRS = 0, ND
 NAGGSRS = 0, NAUTSRS = 0, NRIMRS = 0, NACCRS = 0, NCMELRS = 0, NWETGRS = 0, NDRYGRS = 0 /
&NAM_BU_RRG  LBU_RRG = F, NASSERG = 1, NADVXRG = 0, NADVYRG = 0, NADVZRG = 0, NFRCRG = 0, NDIFRG = 0, NSEDIRG = 0, ND
 NRIMRG = 0, NACCRG = 0, NCMELRG = 0, NCFRZRG = 0, NWETGRG = 0, NDRYGRG = 0, NGMLTRG = 0 /
&NAM_BU_RRH  LBU_RRH = F, NASSERH = 1, NADVXRH = 0, NADVYRH = 0, NADVZRH = 0, NFRCRH = 0, NDIFRH = 0 /
&NAM_BU_RSV  LBU_RSV = F, NASSESV = 1, NADVXSV = 0, NADVYSV = 0, NADVZSV = 0, NFRCSV = 0, NDIFSV = 0, NVTURBSV = 0, N
&NAM_LES  NMODNBR_LES = 1, LTURB_LES = F, NT_LES = 1000 /
&NAM_BLANK  XDUMMY1 = 1.E-2, XDUMMY2 = 0.E+0, XDUMMY3 = 4000., XDUMMY4 = -11., XDUMMY5 = 0.3000000000000007, XDUMMY6
 XDUMMY7 = 0.E+0, XDUMMY8 = 0.E+0, NDUMMY1 = 0, NDUMMY2 = 0, NDUMMY3 = 0, NDUMMY4 = 0, NDUMMY5 = 0, NDUMMY6 = 0, NDUM
 NDUMMY8 = 0, LDUMMY1 = T, LDUMMY2 = T, LDUMMY3 = T, LDUMMY4 = T, LDUMMY5 = T, LDUMMY6 = T, LDUMMY7 = T, LDUMMY8 = T,
                                                                                ", CDUMMY2 = "
                                      ", CDUMMY3 = "
 CDUMMY4 = "                                                                                ", CDUMMY5 = "
                                                ", CDUMMY6 = "
             ", CDUMMY7 = "                                                                                ", CDUMMY
                                                                                " /
&NAM_FRC  LGEOST_UV_FRC = F, LGEOST_TH_FRC = F, LADVEC_RV_FRC = F, LVERT_MOTION_FRC = F, LRELAX_THRV_FRC = F, LRELAX_
 XRELAX_TIME_FRC = 10800., XRELAX_HEIGHT_FRC = 0.E+0, CRELAX_HEIGHT_TYPE = "FIXE", LTRANS = T, XUTRANS = -13., XVTRAN
&NAM_CH_SOLVER  CSOLVER = "SIS                                ", NSSA = 0, NSSAINDEX = 1000*0, XRTOL = 1.E-3, XATOL =
0.1000000000000001, NRELAB = 2, NPED = 1, NMAXORD = 5, LPETZLD = T, CMETHOD = "N", CNORM = "A", NTRACE = 0, XALPHA =
100., XFAST = 0.1000000000000001, NQSSAITER = 1, XDTMIN = 0.1000000000000001, XDTMAX = 600., XDTFIRST = 10. /
&NAM_PARAM_ICE  LWARM = T, CPRISTINE_ICE = "PLAT", CHEVRIMED_ICE = "GRAU" /
```

8.2.2 The diachronic file

It is a file obtained during a segment of simulation or resulting of the conversion of a synchronous file with `conv2dia` for graphical purposes.

The file directly obtained during the simulation has a name ended by `.000`, and contains records such as averaged variables, tendancies, fluxes stored at different times of the simulation on the whole or some parts of the domain. Such records are obtained by asking for temporal series (4.2.27), budgets (4.2.29), aircraft or balloon (4.2.25), profiler or station (4.2.26), LES diagnostics (4.3).

See the documentation: "CREATION et EXPLOITATION de FICHIERS DIACHRONIQUES" (J. Duron, oct. 2001) for more details.

8.2.3 The physiographic file

It is a bidimensional MesoNH file with contains surface datas as orography, vegetation classes, chemical emissions, etc.

See the documentation :"THE EXTERNALIZED SURFACE USER'S GUIDE" for more details.

8.3 References

J. Clochard, 1989: Logiciel de Fichiers Indexés. Direction de la Mtorologie Nationale. Note de travail ARPEGE n°12.

J. Clochard, 1991: Logiciel de Fichiers Indexés. Direction de la Mtorologie Nationale. Technical report.

D. Gazen, 1999: Parallel IO routines. Man page on Meso-NH web site.

C. Fischer, 1994: File structure and content in the Meso-NH model. Meso-NH note.

Appendix A

An illustrative example of use of procedures

A.1 Example of source manipulation

One classical way to obtain your user library is to copy the Master library in your own directory, but you can also create a new library containing only supplementary (compared to the Master library) sources.

For example: a user has a source file "source.f90" which he wants to manage with the Meso-NH macros. First, he creates his RCS library with the name MASTER for the first release of this library:

```
nh-create source.f90 MYOWNL
```

version 1.1 of source.f90 associated to MYOWNL

```
nh-extract source.f90 MYOWNL
```

version 1.1 extracted for modification, next version will be 1.2. After this extraction, you have a file source.f90 which contains the source corresponding to version 1.1. Therefore, you can edit this file and modify it as you want.

```
nh-register source.f90 MYOWNL
```

version 1.2 created and associated to MYOWNL. Note that the file source.f90 which appeared in your directory when the nh-extract was performed, has now disappeared.

From now on, if the user needs to look at the version 1.1 without requiring any modifications, he must do :

```
nh-takeout source.f90 MYOWNL 1.1
```

But let's let the user go on making a new version 1.3:

```
nh-extract source.f90 MYOWNL
```

Therefore, he extracts the last available version i.e. version 1.2, and by modifying source.f90 with the editor, he makes the new version 1.3, registered by the following nh-register operation:

```
nh-register source.f90 MYOWNL
```

However, the user now recognizes an error and needs to remove completly version 1.3, in this case, he must do :

```
nh-unregister source.f90 MYOWNL
```

version 1.2 is now again the (last) default version for MYOWNL.

The user can also create a new branch starting from 1.1, that will be parallel to the main trunk:

```
nh-extract source.f90 MYOWNL 1.1
```

where the new version to be generated will be 1.1.1.1. If, after careful thoughts, the user finally decides not to perform the new release he intended to do when making `nh-extract` , he can undo this extraction:

```
nh-unextract source.f90 MYOWNL
```

In this case, no advance is performed in the release numbers. The Figure A.1 presents a picture illustrating all the actions that have been presented above.

A.2 Example of the prepsource script

Let us now describe a typical interactive `prepsource` session (i.e. with ENVIRONMENT = INTERACTIVE). Below, a succession of steps are presented, in which a user has used the full possibilities of `prepsource`. Lines preceded by a number are output from `prepsource`, whereas other lines correspond to the user's answers:

```
1>please enter a name for your simulation directory
‘pwd‘
2>.............  FULL SCREEN VISUALIZATION OF THE CONTROL PARAMETERS
.................. the user can again proceed to some modifications
3>================================
3>no user library  was specified :
3>Carriage Return alone   to valid
3>something else and <CR> to stop
3>================================
CR
4>/users/come/fischer/mnh_master0/make_mnh.prov does not exist: default is searche
4>d
4>enter the root directory from which source files will be extracted
4>default:
4> 1st :
4> 2nd : /users/come/fischer/mnh_master0
4> 3rd : /mesonh/sources
4>?
$HOME/sources/modif
5>please enter arguments for nh-takeout: file-name version-name [VID] ?
mod_dein.f90 MASTER0
6>1.1 retrieved
7>do you want another file: y or n ? (default y)

8>enter the root directory from which source files will be extracted
8>default:
8> 1st : /users/come/fischer/sources/modif
8> 2nd : /users/come/fischer/mnh_master0
8> 3rd : /mesonh/sources
8>?

9>please enter arguments for nh-takeout: file-name version-name [VID] ?
```

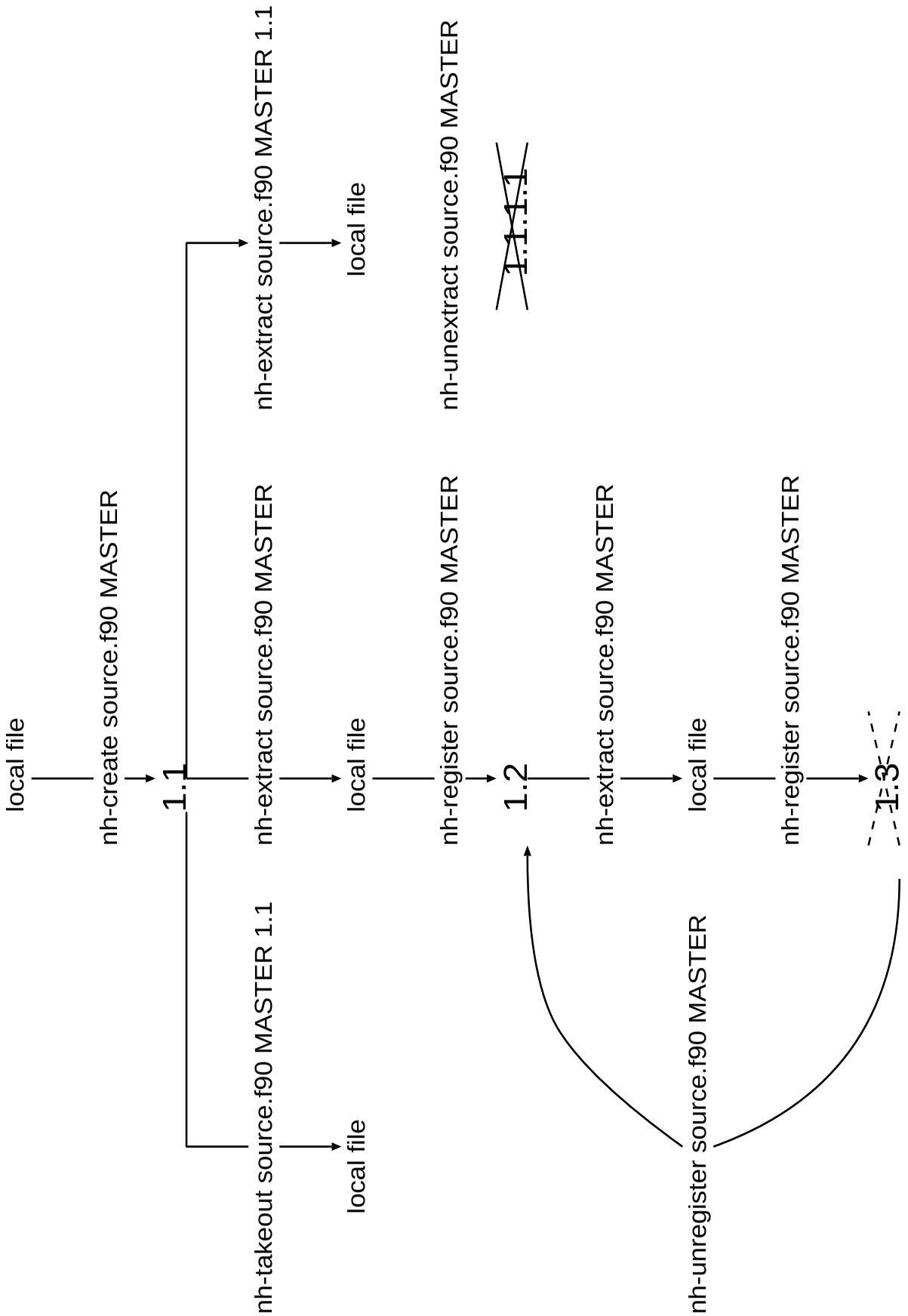


Figure A.1: Example of RCS source file evolution with Meso-NH macros

```
init.f90 MASTER0
10>init.f90 already exists on /users/come/fischer/mnh_master0
10>do you want to overwrite it ?: y or n (default n)
n
11>do you want another file: y or n ? (default y)
n
12>enter the global parameters for compilation on remote machine  (makefile)
12>compilation options ? (default -c)

13>==== end of (silent)interactive prepsource loop ====
13>==== a new file inprepsource.new is now available ==
13>==== for future work with prepsource in SILENTINTERACTIVE
13>mod_dei_n.f90
13>init.f90
```

In steps 1 to 6, the user simply takes out from a RCS library the file "mod_dein.f90". In steps 7 to 11, he asks for a second file "init.f90", which already existed in $ SIMUL.

At step 1, `mesonh` asks for the simulation directory SIMUL; this is because the user did not initialize SIMUL before starting his session. If SIMUL had been initialized before, `mesonh` would not have asked for it.

In step 3, `prepsource` indicates that no name was entered for the user binary library (because null character or 0 was entered). This can be perfectly correct, for example for administrator's tasks or for users who only want to work with the master binary library. In our example, the user simply hit CR to validate and go on.

The first messages in step 4 shows the user that `prepsource` has taken the default makefile for future compilations, because none was existing in the user's SIMUL directory. Then, `prepsource` asks for a directory where the source files have to be extracted from. Two default values are already set up: the second corresponds to the DIRUSER catalog read in prepsourcerc, the third to the DIRMASTER catalog (which is global to Meso-NH). The first default will only be set up after the first source file will have been extracted (see step 8). The user here enters another directory name, in which `prepsource` will first check for the source file; however, if it does not find anything, it will go on searching in the defaults. The file name and version name are given in step 5, just in the way they would be given on the `nh-takeout` command line. We stress *that* `mesonh` *does not perform any RCS manipulation.*Thus, the modified file is purely local and not reintroduced into its home RCS library by `prepsource` . Moreover, it will be deleted from the local directory at the end of prepsource.

From step 7 onwards, the sequence is basically the same than the one described above, except for minor changes. In step 8, the first default is now set up to the directory name in which `prepsource` has found the first source file. In step 10, `prepsource` recognizes that the source file init.f90 already exists on $SIMUL: it thus asks whether the file has to be overwritten (this means, reextracted from RCS) or not.

In step 11, the user stops the general `prepsource` loop.

He finally enters the compilation options for the remote machine. Step 13 consists of some information kindly printed by `mesonh` . It indicates what source files are prepared for remote compilation (here, finally, two source files which contained multi-task code and which were customed for NSOURCE=1) and it says that a file inprepsource.new was created, which can be used for a future SILENTINTERACTIVE session with `prepsource` .

Be careful when handling modules! If the user tries to change modules that exist in the master library, `prepsource` will exit. Indeed, modifications in the MODD_ and MODN_ modules from the master RCS library are prohibitted.

Let us now consider the case where ENVIRONMENT = SILENTINTERACTIVE. Then, `mesonh` is not asking anything to the user (that's why it is silent) but it still prints information on the screen (so it keeps some interactive shape). In fact, `prepsource` is now reading input from a file that must be called "inprepsource" and be present in the catalog $SIMUL. This file contains the following information (see the `prepsource` man page): dir1, file1 VERSION, dir2, file2 VERSION, dir3, file3 VERSION, ... ,–, compiler options. Each information separated by commas is stored on one line. Thus, most part of inprepsource consists of pairs of lines containing first a directory name and second a file and version name. This is in fact the input that `prepsource` asked for in steps 4 and 5 of the previous example. The last two lines contain a separator string '–', followed by the compiler options (which were asked for in step 12 above). Such a file inprepsource can of course be directly written by the user. However, `prepsource` itself generates during each session a file called "inprepsource.new" that may be used in a future session after being copied to "inprepsource".

When ENVIRONMENT = BATCH, `prepsource` is running without any contact to the user. In this case, output is directed onto a temporary file and no input is read at all. `mesonh` simply takes all source files that are stored in $SIMUL and prepares them for remote compilation. Thus, the user could have worked interactively before this batch session, and then left his source files on $SIMUL and run `mesonh` in batch.

Some specific remarks concerning `prepsource` are now made. We recall that the input parameters (present in prepsourcerc) concerning the user's objects (BIBUSER, etc ...) all are relative to the $HOME catalog.

A.3 Example of the prepmodel script

Let us now describe a typical `prepmodel` session (with ENVIRONMENT = INTERACTIVE). As for `prepsource` , the `prepmodel` output is preceded by a line number, whereas the user's answers are not:

```
1>.................. FULL SCREEN VISUALISATION OF THE CONTROL PARAMETERS
.................. the user can change them interactively here again
2>=========================================
2>prepmodel:no user library  was specified :
2>        Carriage Return alone   to continue
2>        something else and  CR  to stop
2>=========================================
CR
3>========= no user library used
3>NMODEL 2
4>.............. an edition of the files' get commands is proposed
................ the user may change them
5>============================================================
5>your job is ready to execute the model(s) :
5> execute it alone by "tosupc /users/come/fischer/mnh_master/outprepmodel"
```

```
5>       or
5> add it to /users/come/fischer/mnh_master/outprepsource
5> before executing the global job by tosupc
5>============================================================
```

Here step 1 permits the user to change on the screen the control parameters (the user moves from one field to another using the arrow keys; once a field is changed, it is validated by a CR; to leave the page and to go on, the space key has to be pressed).

Step 2 is equivalent to `prepsource` step 3: `prepmodel` is simply concerned about the user not having indicated any user library name (but 0). This may not be an error, if '0' is retained, this simply means that all the compiled object code will be extracted from the master binary library. Of course, for the general use and task during development or diagnostic investigation with the Meso-NH code, the user will indicate his own binary library.

Step 4 allows the user to verify and modify the command lines used to get the input files (a file containing this code is edited) because GETCONTROL=yes. This could prove helpful if special files have to be got.

In cases where ENVIRONMENT = SILENTINTERACTIVE or BATCH, `mesonh` will not ask anything to the user nor read in any file (a file inprepmodel does not exist). In SILENTINTERACTIVE, `prepmodel` will still print output information, whereas it is dumb in BATCH.

Finally, let us emphasize some specific points. First, if the control variable MAINPROG is set to 'MODEL', `mesonh` will try to read some information (the name of the Meso-NH input data files, the number of models) in the namelist formatted file EXSEG1.nam (and EXSEG2.nam etc... in multitask). Thus, this file has absolutly to be present on the $ SIMUL directory. If MAINPROG is set to 'PREP_IDEAL_CASE', then a namelist called EXPRE1.nam is read. Again, make sure in this case that this file exists and is up to date on $ SIMUL.

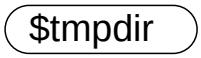
One of the main jobs in `prepmodel` is to get and save the Meso-NH input and output files on the remote machine on which the job will be executed. Thus, there is some moving around of information and you have to be aware of it. Furthermore, for some specific use like debugging, you need to know where you can find all you need to run `totalview`. Figure A.2 gives an overview of what kind of files are copied and where. Mainly, it should be recalled that the code is compiled on $TEMP, that it is run on $TEMP/execdir, and that the source, as well as the executable code is saved on $ workdir if DEBUG = cdbx.

We stress that the Meso-NH output data files are stored via the script `transfer.x` , which is called directly by the fortran programs. However, you need to define carefully the parameters OUTHOST and OUTDIR, to make sure that `transfer.x` is saving the files where you expect it. We recall also that depending on the dummy argument KTYPE in FMOPEN, the descriptive and LFI part of a Meso-NH file will be concatenated via `cpio` or not. If file transfers have failed, `prepmodel` will try to save the local data files either on SuperCalc (if enough space) or on Archive.

A.4 Example of the tosupc script

Let us now look at a typical INTERACTIVE `tosupc` session (conventions are the same as in the previous examples).

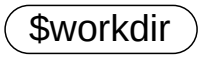
```
1>.............. FULL SCREEN VISUALISATION OF THE CONTROL PARAMETERS
.............. the user can change them here
1>------------- tosupc script
```



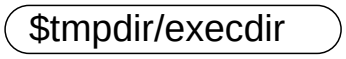
copy or creation:

*.f90, *.o, *.l
BIBUSER
necessary shell scripts

ABSMOD.exe
namelists
input FM-files



if DEBUG=cdbx:
copy: namelists, *.f90, input FM-files, ABSMOD.exe
this allows interactive debugging on workdir



copy: namelists, input FM-files, ABSMOD.exe
after execution (ABSMOD.exe run):
ouput FM-files are stored on Supercalc
($HOME or $workdir),
otherwise on Archive and on $workdir/$OUTDIR

Figure A.2: The objects that are copied or created on temporary and work directories during a Meso-NH run

```
1>simulation directory SIMUL =/users/come/fischer/mnh_master
2>TOSC toto XX NORM 1200 384 1 no 1
2>SUPER-CALCULATEUR D EXECUTION ? :
2>----------------------------
2>passage par xoper NQS vers
2>  tora ou toba     pour le VPP5000
2>  gmor    pour le VPP5000 - class getmor
2>  gmax    pour le VPP5000 - class getmax
2>  toco    pour le VPP5000 - class compil
2> pour les super-calculateurs non meteo :
2>  idris    pour l'idris : passage par hebe
2>  par defaut: tora
kami
```

Step 1 gives the possibility to the user to change the control parameters, which can however also be initialized directly in file execfilerc. In step 2, you have to enter the name of the super computer in which the job will be executed.

In SILENTINTERACTIVE or BATCH, `tosupc` will ask nothing to the user but simply use the information in `tosupcrc` to work. Also, in both modes, `tosupc` will not ask for the NQS parameters and you thus have to initialize them correctly in the `tosupcrc` file (an example of such a file is stored on $ MESONH/procedures).

Index